공학박사 학위논문

Synthesis of Low Noise All Digital PLL

저 잡음 디지털 위상동기루프의 합성

2014 년 2 월

서울대학교 대학원

전기.정보공학부

김 우 석

Ph.D. Dissertation

Synthesis of Low Noise All Digital PLL

저 잡음 디지털 위상동기루프의 합성

By

WooSeok Kim

February 2014

Department of Electrical and Computer Engineering
College of Engineering
Graduate School of Seoul National University

Synthesis of Low Noise All Digital PLL

지도 교수 정덕균

이 논문을 공학박사 학위논문으로 제출함
2014 년 2 월

서울대학교 대학원

전기.정보공학부

김 우 석

김우석의 공학박사 학위논문을 인준함
2014 년 2 월

위 원 장 김 수환 (인)

부위원장 정 덕균 (인)

위 원 김 재하 (인)

위 원 박 재진 (인)

위 원 김태익 (인)

Abstract

As a device scaling proceeds, Charge Pump PLL has been confronted by many design challenges. Especially, a leakage current in loop filter and reduced dynamic range due to a lower operating voltage make it difficult to adopt a conventional analog PLL architecture for a highly scaled technology. To solve these issues, All Digital PLL (ADPLL) has been widely studied recently. ADPLL mitigates a filter leakage and a reduced dynamic range issues by replacing the analog circuits with digital ones. However, it is still difficult to get a low jitter under low supply voltage. In this thesis, we propose a dual loop architecture to achieve a low jitter even with a low supply voltage. And bottom-up based multi-step TDC and DCO are proposed to meet both fine resolution and wide operation range. In the aspect of design methodology, ADPLL has relied on a full custom design method although ADPLL is fully described in HDL (Hardware Description Language). We propose a new cell based layout technique to automatically synthesize the whole circuit and layout. The test chip has no linearity degradation although it is fully synthesized using a commercially available auto P&R tool. We has implemented an all digital pixel clock generator using the proposed dual loop architecture and the cell based layout technique. The entire circuit is automatically synthesized using 28nm CMOS technology. And s-domain linear model is utilized to optimize the jitter of the dual-loop PLL. Test chip occupies 0.032mm2, and achieves a 15ps_rms integrated jitter although it has extremely low input reference clock of 100 kHz. The whole circuit operates at 1.0V and consumes only 3.1mW.

Keywords : PLL, Cell Based, Synthesis, Jitter, Pixel Clock, Dual Loop.
Student Number : 2010-30216

Contents

Lists of Figures

Lists of Tables

1. Introduction

1.1 Thesis Motivation and Organization

1.1.1 Motivation

The CMOS scaling has driven the growth of semiconductor industry by achieving a higher performance with less cost. And the industry has continued the CMOS scaling to get a further profits. This kind of virtuous cycle has leaded the semiconductor industry since the early 70's, and it has been proved its effectiveness. This trend is the well-known as "Moore's Law"[1]. However, the scaling cannot be an all-mighty solution anymore as the more mixed analog blocks are integrated on the same chip with conventional digital circuits. Although the scaling provides advantages such as small size and high speed device, it also degrades the some device parameters which are essential for high performance analog circuit [2].

In this work, we introduce a general scaling theory and the following challenges in designing a Phase Lock Loop (PLL) clock synthesizer. In order to solve the design issues, we suggest a new design methodology and PLL circuit architecture which are very friendly to a nanoscaled CMOS technology.

1.1.2 **Thesis Organization**

In chapter 1, we introduce design challenges due to a highly scaled technology.

In Chapter 2, the basic theory of the conventional charge pump PLL and the ADPLL are covered. In this part, we provide a noise analysis and a jitter optimization theory too.

Chapter 3 suggests a new All Digital PLL (ADPLL) architecture having low jitter, small size and low power. The PLL is fully described in the Hardware Description Language (HDL) and synthesized automatically using an auto P&R tool. To avoid linearity degradation during auto P&R process, the new cell based design methodology is suggested. To verify the proposed ADPLL architecture and design methodology, a prototype chip has been realized using a standard 28nm CMOS technology. The measurement results show that it consumes only 0.032mm^2 areas and 3.1mW power at 1.0V operating voltage. In addition, the dual loop architecture satisfies a small integrated jitter (15ps_rms) under the extremely low input frequency (100kHz) and loop bandwidth (10kHz). The synthesized ring oscillator DCO shows a good linearity performance which is comparable to the manually drawn DCO.

1.2 PLL Design Issues in Scaled CMOS Technology

In this chapter, we'll show the design challenges and remedies to solve the problems. The PLL Design challenges in a nano-scaled CMOS technology are illustrated in Fig. 1.1.

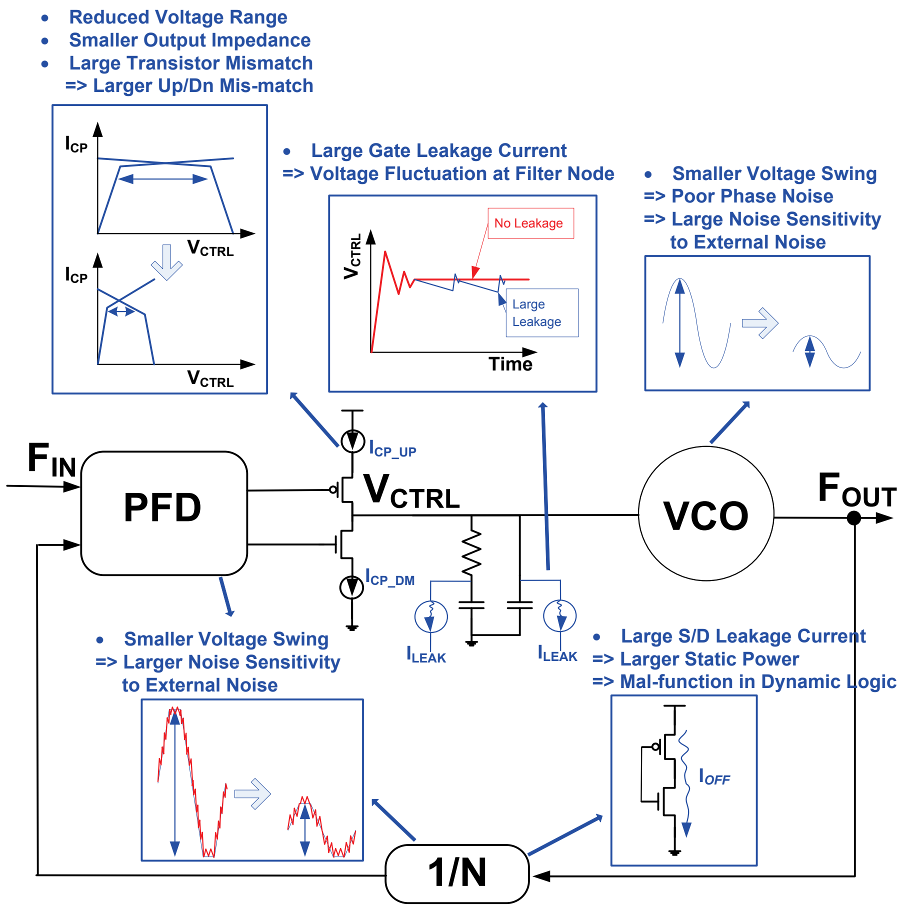


Fig. 1.1 PLL Design Challenges in Highly Scaled MOSFET Technology.

1.2.1 **Low Supply Voltage**

As the devices are scaled down, the supply voltage should be reduced to guarantee a constant electric field. However, a lower supply voltage degrades the phase noise of internal oscillator by limiting the voltage swing. The Hajimiri's work [3] shows that this fundamental limitation in the voltage swing sets a limit in the phase noise of oscillator. From the Navid *et al* [4], a minimum achievable phase noise of a ring oscillator is represented by (1.1), where offset frequency Δf, oscillation frequency f_O, Boltzman constant k, temperature T, the number of ring stages N, loading capacitance C, and voltage swing v_{sw}.

$$PN_{MIN}(\Delta f) \approx \frac{7.33 \cdot f_O \cdot k \cdot T}{N \cdot C \cdot v^2_{sw} \cdot (\Delta f)^2} \tag{1.1}$$

The maximum value of v_{sw} is determined by a supply voltage. If the supply voltage swing is not large enough then more current consumption is required to generate the same oscillation frequency with larger N and C.

The low supply voltage also reduces a dynamic range of VCO control voltage. It means the VCO gain should be larger for a same frequency tuning ranges. The larger VCO gain degrades a jitter because the VCO output is more easily modulated by a noise in the control node. The available VCO tuning range is additionally narrowed due to a limited dynamic range of the charge pump. The dynamic range of the charge pump is determined by the voltage range satisfying a reasonable UP/DN pump current matching. Unfortunately, this range is generally less than 30% of the supply voltage in the nano-scale/low voltage process. A cascode current mirror cannot be utilized because the voltage headroom is not enough.

A dual voltage architecture is utilized to obtain an enough voltage headroom and large voltage swing [5]. In this architecture, the noise sensitive analog blocks (VCO, Charge Pump, PFD) are implemented with thick gate oxide transistors operating at a high supply voltage (1.8V ~ 3.3V). And the remaining high speed digital blocks utilize a thin gate oxide transistors running in a low supply voltage. The multi voltage domain architecture achieves a wide dynamic range and low noise by sacrificing the size and power.

Another design challenges due to the lower supply voltage is that a circuit is more susceptible to an external noise such as supply and substrate noise. If noise amplitude is same then a SNR (Signal to Noise Ratio) is proportional to the amplitude of an original signal. That is, the effect of an external noise increases as the supply voltage is scaled down. In addition, the operating speed generally becomes higher as the technology scaling proceeds. Thus, it makes a larger switching noise. Therefore, the switching noise degrades a circuit performance more severely in a highly scaled technology.

While the dual voltage architecture can mitigate a supply noise by adopting a cascode scheme, but this is not enough in a highly noisy environment. Many studies have been executed to find the ways to reduce the noise effects [6-16]. Most widely used technique is to regulate a noisy supply before providing for a noise susceptible analog block [7, 8, 10, 13]. In terms of size, the regulator needs a large size of decoupling cap connected to load site. And the amplifier in a linear regulator should be fast to filter out a high frequency noise component. Practically, the regulator loop bandwdith should be larger than the closed loop bandwidth of the PLL to keep the PLL loop being stable. And the regulator drop voltage should be minimized to allow a large voltage swing in the VCO oscillation nodes. Although the supply regulation is helpful to mitigate the noise, it requires an additional size, power and extra power supply source.

To overcome this problem, noise cancellation techniques have been studied. The basic idea is to cancel out the effect of a supply noise by summing a negative and positive terms [9, 11, 14, 16]. The effect of a supply noise is removed by adding another compensation signal having a reverse polarity. The amplitude of compensation signal should be the same with the injected noise signal and have a reverse polarity. In order to achieve a perfect cancellation, a back ground calibration scheme is utilized which defines the proper amplitude of the compensation signal according to an injected noise amplitude; which is not fixed value but varies according to the chip operating mode and a PVT condition.

1.2.2 **High Leakage Current**

As the process scaling proceeds, it's more difficult to cut off a leakage current path completely. There are two leakage paths which are gate-tunneling and source-drain leakage. The gate tunneling current increases because the gate oxide thickness decreases in a highly scaled technology [17]. This makes it difficult to use a MOS capacitor as a loop filter component. The loop filter stays at a floating state during most of the time after the PLL is locked. If there is a large gate leakage then a stored charge is leaked and the node voltage of a loop filter is changed. These periodical fluctuations degrade a jitter value by modulating a VCO output period. There are fancy leakage current compensation technologies implemented with an analog circuit technology [18, 19]. And leakage free capacitor such as inter-metal capacitor and thick gate oxide capacitors are widely used for loop filters. Especially, the inter-metal capacitor can be a good alternative in a highly scaled technology because the capacitance per area is comparable to the oxide capacitance; the capacitance is inversely proportional to the distance between metal electrodes (Fig.A.25).

The leakage current from drain to source are problem. As the transistor channel length decreases, it's more difficult turn off a transistor completely because the sub-threshold slope is generally not scaled. And the short channel effects such as DIBL (Drain Induced Barrier Lowering) contribute to the source/drain leakage current by decreasing the effective threshold voltage of a transistor [17]. The leakage current from drain to source increases a static power increase, especially this is becoming a big problem as the number of transistors increases. In the aspect of circuit operation, the high source/drain leakage current results in a circuit failure in a dynamic logic circuit such as

TSPC (True Single Phase Clock) logic by sinking the charges stored at a floating node [20]. This charge leakage during the evaluation phase limits the minimum operating speed of a dynamic logic circuit.

1.2.3 **Device Reliability: NBTI, HCI, TDDB, EM**

As the device size shrinks, it's unavoidable that a higher electric field applied within a device. While the higher electric field is helpful to achieve higher speed by accelerating the carriers, it's harmful for device reliability. In this section, we'll briefly cover the reliability issues confronted in a scaled technology. Fig. 1.2 shows the HCI (Hot Carrier Injection) phenomenon. The constant field scaling cannot be applied in real world because there is a limitation in scaling supply voltage (Table.A.1 and Table.A.2). It means that the intensity of the electric filed between source and drain becomes stronger as the scaling proceeds. An electron having a large kinetic energy, which is provided by a strong electric filed, are moved on to the silicon oxide overcoming the energy barrier. This hot carrier makes some defects at silicon dioxide interface and degrades a device performance.

Fig. 1.3 illustrates the NBTI (Negative Biased Temperature Instability) effects. The hole is attracted to oxide interface by a strong vertical field between gate and channel and it results in defects at the interface between gate oxide and channel, which becomes severe at a high temperaturet. The NBTI is occurs only in PMOS transistor, there is counterpart phenomenon called PBTI which is for NMOS transistor, but it has less effects compared to the NBTI. The NBTI and HCI increase a threshold voltage and reduce a transconductance and a saturation current, which is presented in Fig. 1.4

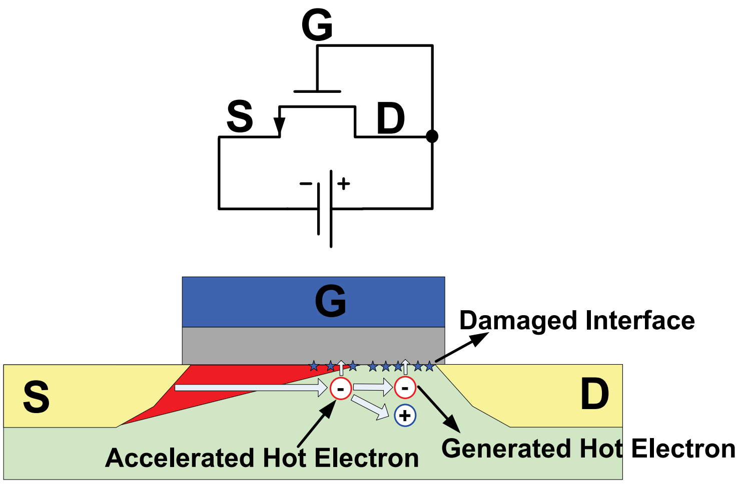


Fig. 1.2 HCI (Hot Carrier Injection) Mechanism

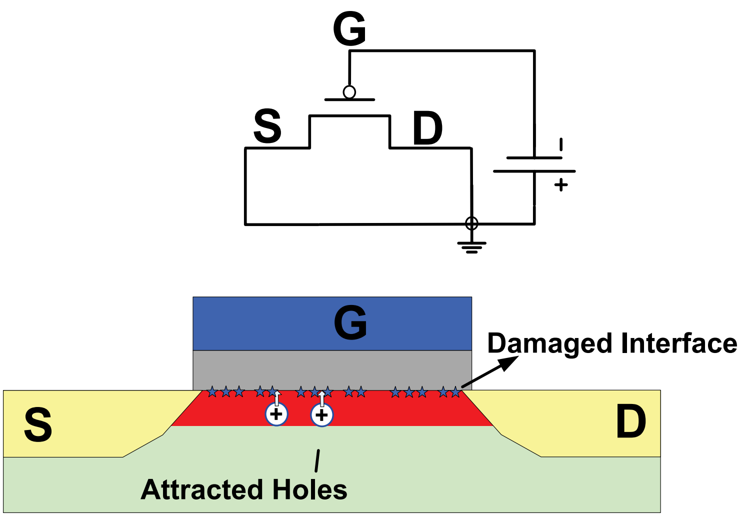


Fig. 1.3 NBTI (Negative Biased Temperature Instability) Mechanism

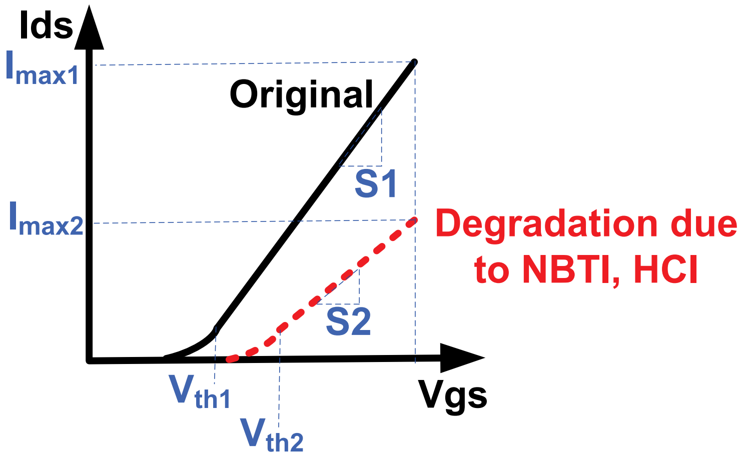


Fig. 1.4 I/V Characteristic change due to NBTI and HCI.

While the HCI and NBTI only degrade the device performances, the TDDB (Time Dependent Dielectric Breakdown) and EM (Electro Migration) result in a catastrophic disaster. Fig. 1.5 shows the TDDB failure. If a large voltage is applied across the gate oxide then numerous defect are generated. Finally, the oxide is broken down when the applied voltage exceeds a maximum allowance and a current path is generated along the generated defects. The broken oxide doesn't work as an insulator anymore and the transistor is destroyed permanently.

The Electro Migration occurs when the current density is too high. As the high energy electrons collide with the atom, the atom particles are also moved onto a positive electrode. As depicted in Fig. 1.6 , the more electrons hit the copper atom and the metal line is opened in the middle and the end region is bulged and shorted to the near metal as the atoms are moved and accumulated in the end region. To prevent EM failure, the current density should be kept low by increasing the cross section area of a metal wire.

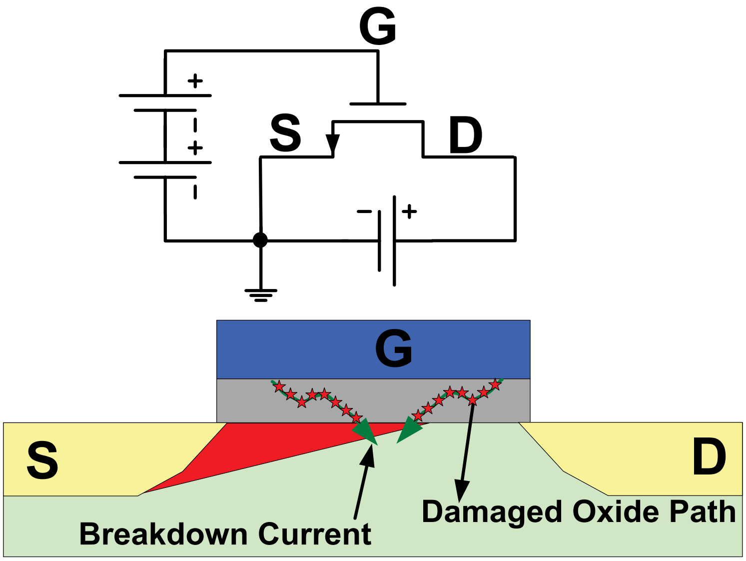


Fig. 1.5 TDDB (Time Dependent Dielectric Breakdown) Mechanism

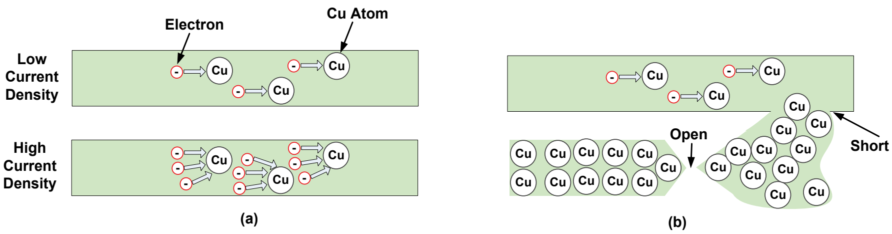


Fig. 1.6 Electro Migration Failure. (a) Before. (b) After being damaged

1.2.4 Mismatch due to Proximity Effects: WPE, STI

As the devices are placed more closely, the transistor performance is more easily affected by adjacent patterns. In a nanometer regime, an inaccuracy in patterning technologies such as etching and lithography are reduced but the effect due to a proximity effect increases [2]. The first one is the WPE (Well

Proximity Effect). This phenomenon occurs when a transistor is located closely to a well edge. The photo-resist for pattering a well area reflects dopants for a well region, and the area being close to the well edge is more highly doped than expected. In conclusion, WPE changes the threshold voltage of the transistor; V*th* increases due to the higher doping concentration [17]. Fig. 1.7 provides the WPE mechanism, the implanted dopants for the well region is scattered by photoresist wall and the reflected dopants penetrate into the adjacent active device region. The dopant type for the well is the same with the one implanted for a channel, therefore the threshold voltage increases due to higher doping concentration, which is higher than originally targeted value. The effect is inversely proportional to the distance between well edge and active transistor area as shown in Fig. 1.7. To mitigate WPE, the active device should be placed at a long distance (> 1~3um) from the edge of well [21].

The second proximity effect is a mechanical stress induced by STI (Shallow Trench Isolation). When a transistor is located closely to the STI, a mechanical stress is induced and this force changes the lattice structure of a channel region. Finally, the distortion in lattice structure causes the changes in mobility, threshold voltage, and saturation current. The changes of device parameter are proportional to the intensity of mechanical stress which is inversely proportional to the distance between transistor and STI region (Fig. 1.8) [22]. While the WPE can be completely removed by placing a transistor with a long distance from a well edge, but the STI effect cannot be disappeared because the distance between STI and active area is automatically determined by the end of active device. However, the STI effect can be relaxed by inserting a dummy pattern between real device and STI region [21, 22].

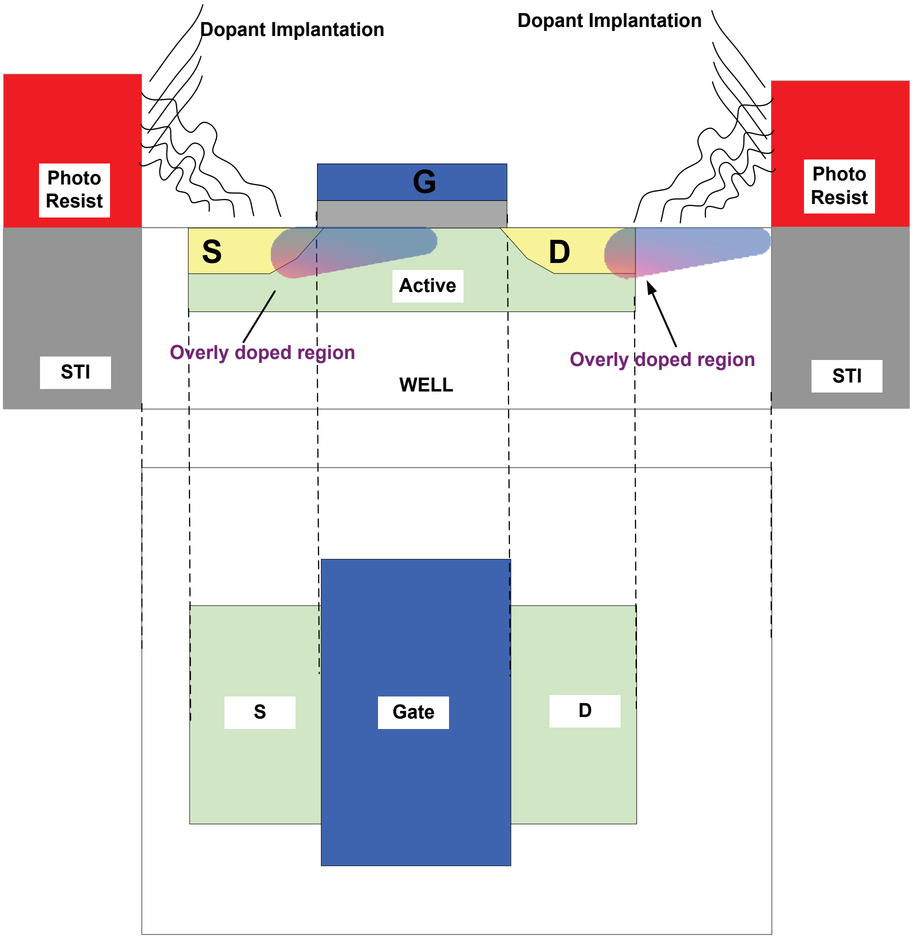


Fig. 1.7 WPE (Well Proximity Effect)

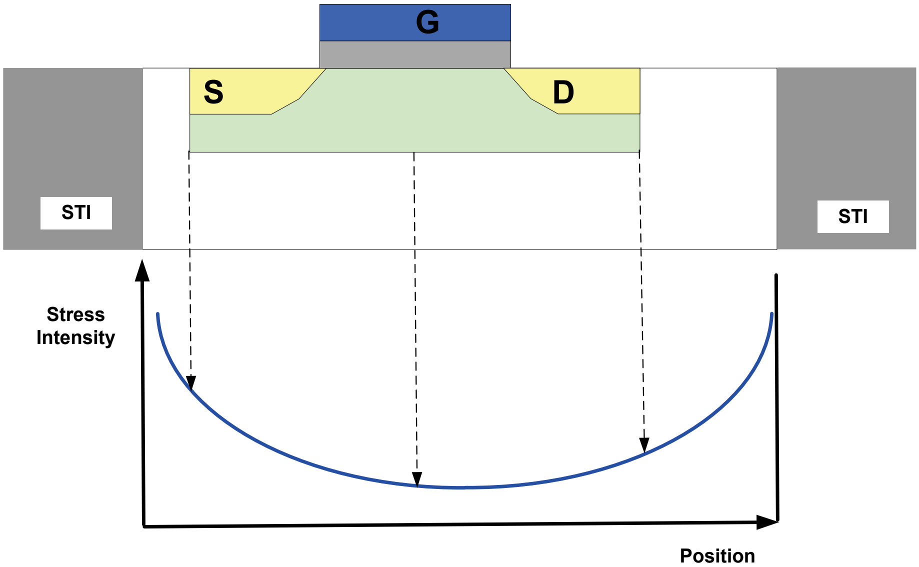


Fig. 1.8 The profile of the mechanical stress due to STI effect.

1.3 Overview of Clock Synthesizers

In this section, we'll overview the prior clock synthesizers to overcome the design challenges in scaled CMOS technology. The solutions focus on achieving a low phase noise under a low supply voltage, and reduce the effects of leakage current in loop filter.

1.3.1 Dual Voltage Charge Pump PLL

The conventional, charge pump PLL has been widely used due to its simplicity and good performance. However, a thin gate oxide transistor cannot be used for the loop filter due to a large leakage current, and it becomes more difficult to achieve an acceptable phase noise under low supply voltage [3, 4]. If one placed the think gate oxide transistors with thick gate oxide transistors then this problem would be solved. However, a thick gate oxide transistor is slower and consumes more power than a thin gate oxide transistor.

To mitigate the speed degradation and power consumption, the both transistors can be used on the same chip as shown in Fig. 1.9 [5].

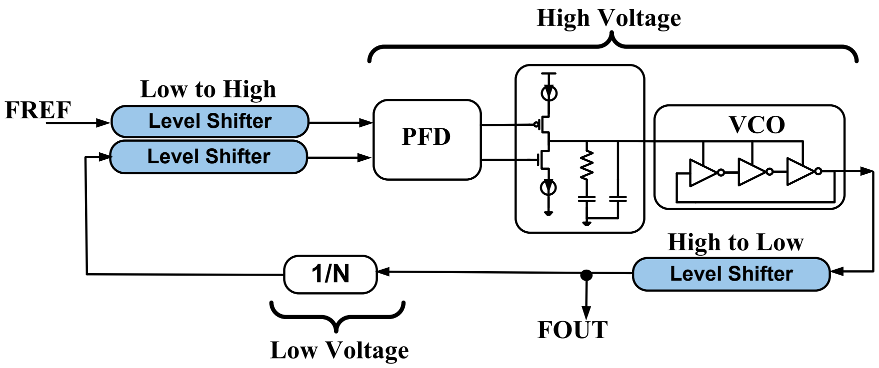


Fig. 1.9. Dual Voltage Charge Pump PLL

In this architecture, the noise sensitive analog blocks such as VCO, Charge Pump, Loop filter, and PFD are implemented with thick gate oxide transistors. The loop filter leakage is suppressed by adopting thick gate MOS capacitor. And a VCO achieves a better phase noise because a swing voltage increases. Of course, the VCO needs an additional power to deal with a large swing and slower transistors. Unlike the analog blocks, the high speed digital blocks such as a divider and an output buffer utilize thin transistors. The level shifters should be placed at the voltage domain interfaces. Two low to high level shifters are placed in front of high voltage operating PFD, and high to low level shifter is inserted between VCO and feedback divider. By separating high speed block and low noise block, the power consumption and speed degradation are mitigated.

1.3.2 DLL Based Edge Combining Clock Multiplier

PLL is fundamentally susceptible to the jitter accumulation due to poor phase noise of the VCO. To filter out a jitter accumulation, a larger loop bandwidth is required, but it is limited by the well known stability requirements; a loop bandwidth should be less than 1/10 of the input clock frequency [23]. That is, the input clock frequency should be large enough, but it is not always feasible.

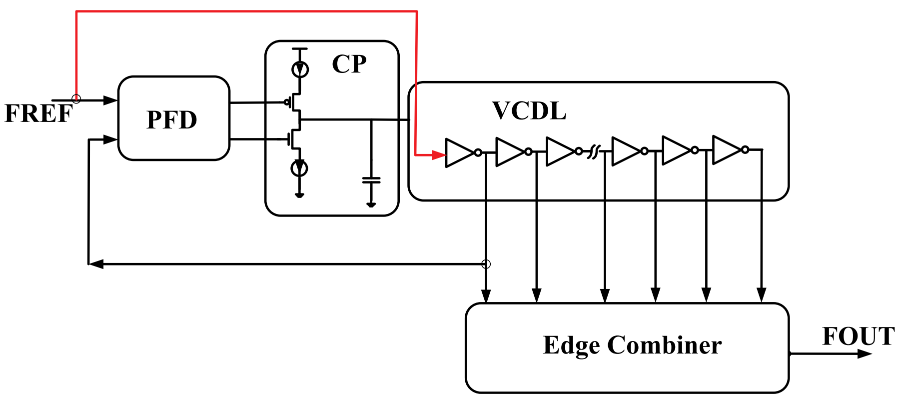


Fig. 1.10. DLL Based Edge Combining Clock Multiplier

Unlike a PLL, a DLL (Delay Locked Loop) does not suffer from jitter accumulation because a output clock is only delayed input signal. To generate a multiplied clock signal, multi-phase signals from VCDL (Voltage Controlled

Delay Line) are processed in edge combiner block (Fig. 1.10) [24]. A jitter performance is dominated by the uniformity of multi-phase signals [25-27] . Compared to a conventional PLL, a edged combiner based clock multiplier is less attractive because it is highly dependent on the process variation and line mismatches in VCDL. According to the prior arts, a conventional PLL is better considering mismatches [24]. In addition, the edge combiner is difficult to have various multiplying factors.

1.3.3 Recirculation DLL

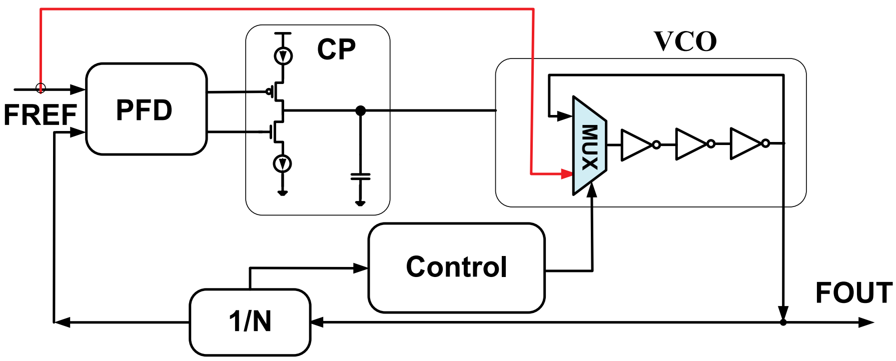


Fig. 1.11. Recirculation DLL

To prevent a jitter degradation due to mismatches, and to achieve a various multiplying factors, the recirculation DLL was proposed [28-35]. As shown in Fig. 1.11, the feedback path of a VCO is opened periodically and the input reference clock FREF is forcibly inserted into VCO. The periodically inserted clean input resets the accumulated phase noise and an in-band noise is filtered out. However, the improvement is not huge unless the input frequency is fast. In addition, the glitch during the MUX switching degrades a jitter performance. Furthermore, if you could use a high frequency input clock then you had better simplify a design by increasing a loop bandwidth.

1.3.4 **Reference Injected PLL**

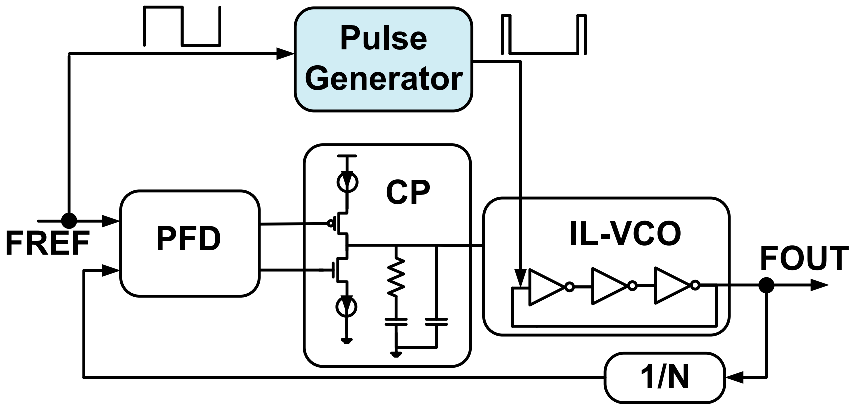


Fig. 1.12. Reference Injected PLL.

Injection locked PLL (Fig. 1.12) is very similar to the recirculating DLL (Fig. 1.11) in the aspect that the input clock signal is used to clean the accumulated phase noise of the VCO. But Reference Injected PLL does not cut off the feedback path of VCO but makes a VCO lock to a input reference signal [36-41]; that is, it has an injection locked VCO. While Recirculation DLL should be implemented with a ring VCO, the reference injected PLL can have a LC VCO either. Anyway this architecture also should have a high frequency input clock to achieve a significant jitter reduction.

1.3.5 **All Digital PLL**

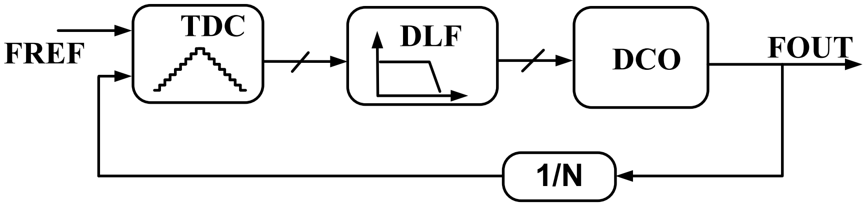


Fig. 1.13. Conventional All Digital PLL.

The architecture suggested from the section 1.3.1 to 1.3.4 are basically charge pump PLL (or DLL) having a loop filter and a charge pump. It means that they would suffer from a leakage current of loop filter, bulky loop filter size, and narrow dynamic range of charge pump. These design issues will become worse as a technology scaling proceeds. Otherwise, a highly scaled technology improves a timing resolution due to its improved operation speed. Fig. 1.13 illustrates a conventional All Digital PLL (ADPLL). The input and output signal of the internal blocks are digital code. Especially, the analog loop filter is replaced with a digital one. There is therefore no leakage problem and dynamic range limitation. In addition, ADPLL is more suitable for highly scaled technology due to following reasons. First, ADPLL is less affected by device parameters such as intrinsic gain, output impedance, and leakage current. Second, the quantization noise is reduced as the device operating speed improves.

In the aspect of design methodology, ADPLL can be described by HDL (Hardware Description Language). This does not only reduce the simulation time but also give a chance to synthesize whole design automatcially.

While the ADPLL has many advantage over a conventional CPPLL (Charge Pump PLL), the quantization noise of TDC and DCO degrade a jitter performance [42]. And a DCO (Digital Controlled Oscillator) has larger amount of noise compared to a VCO, because larger switching operation exists.

1.3.6 **Flying Adder Clock Synthesizer**

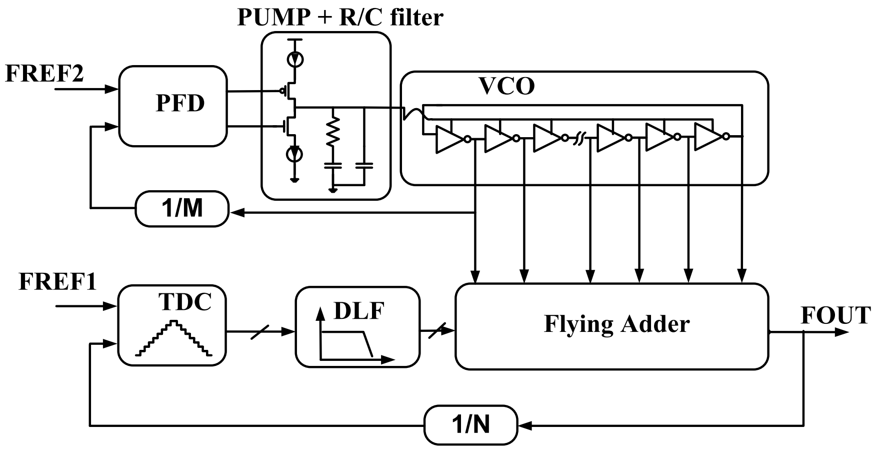


Fig. 1.14. Flying Adder Clock Synthesizer.

Even though there are many techniques to improve a DCO resolution [43, 44]. The phase noise due to flicker and thermal noise is fundamentally dominated by voltage swing and oscillator topology [3, 4, 45]. Like the VCO, the DCO also suffers from phase noise degradation in highly scaled low voltage technology. While the in-band noise coming from a DCO can be filtered out by increasing a PLL loop bandwidth, this measure is only available when the reference clock is high enough; input clock frequency should be higher than 10 times of loop bandwidth. When a reference clock has low frequency, a DCO should have low phase noise to meet a jitter requirement. But a DCO has poor phase noise compared to a VCO because the digital tuning elements work as noise sources; switching noise, flicker noise, and thermal noise.

The flying adder architecture was proposed to increase a PLL loop bandwidth larger than the fundamental limitation [46-49]. It is composed of two PLLs as shown in Fig. 1.14. The main digital loop gets a low frequency clock of FREF1 and synthesizes a target output clock (FOUT) using the flying adder block of the main digital loop. Whereas, the secondary analog loop is conventional charge pump PLL having higher input reference; therefore, the loop bandwidth can be increased larger than the main loop's one. The multiphase clocks from the secondary loop are properly synthesized to generate a target "FOUT". While the proposed architecture help reducing a phase requirement for a internal VCO. The overall jitter is highly dependent on the uniformity of the multi phase clocks. And the secondary loop has the same design challenges of the conventional charge pump PLL.

1.3.7 **Dual Loop Hybrid PLL**

Fig. 1.15 shows another technique to suppress a DCO phase noise. In this architecture, the DCO is implemented using a conventional chare pump fractional-N PLL of which reference clock is from an external crystal oscillator. The output clock of the fraction-N PLL has excellent long-term jitter because the high frequency reference and large loop bandwidth remove a intrinsic noise of VCO. The output frequency of fractional-N PLL is controlled by setting the feedback divider ratio "M", which is controlled by main digital loop operating at a relatively lower frequency. That is, the proposed hybrid PLL has the dual loops composed of the slow digital loop operating in a low frequency (FREF1) and the fast analog loop operating at a crystal oscillator clock frequency (FREF2).

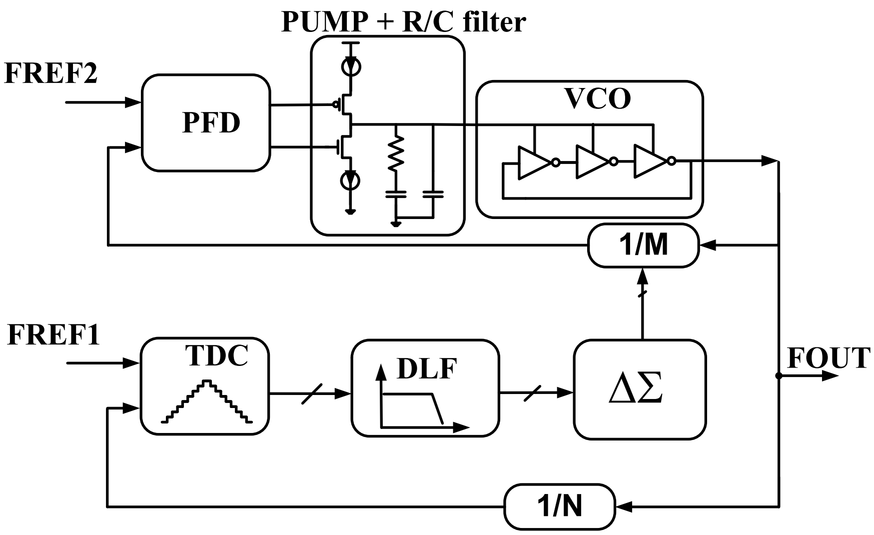


Fig. 1.15. Dual Loop Hybrid PLL.

The fractional-N PLL, which is used as a DCO, has a high loop bandwidth and very clean reference clock, so its phase noise is improved. Eventually the proposed analog/digital hybrid loop PLL has a good phase noise even though the slow loop has a low loop bandwidth. Therefore it can be a good candidate for a clock generator having a low input clock. But it has a limitation that conventional analog PLL is necessary in the fast loop. Eventually the analog/digital hybrid PLL will suffer from same challenges that a conventional charge pump PLL has.

1.3.8 **Comparisons**

Table. 1.1 summarizes the comparison results between the prior arts and this work. In terms of the filter size and the leakage current, the hybrid PLL and the flying adder architecture offer only the limited advantage because these techniques are not fully in the digital domain. They still need a utilized analog PLL to realize the DCO function. Unlike the hybrid PLL and the flying adder, the conventional single-loop ADPLL might solve the leakage and filter size problem, but the DCO noise is still a bottleneck for attaining low jitter. Though the dual-loop architectures such as the hybrid PLL and the flying adder configuration are helpful for reducing the DCO phase noise, power consumption and design complexity are significantly larger than the single-loop architectures. In this work, we propose all digital dual-loop PLL. As shown in Table. 1.1, an all digitalized dual-loop PLL does not suffer from conventional design challenges such as leakage, bulky filter size, and jitter accumulation. In addition, whole design is described in HDL and synthesized using auto P&R tools. Of course, there is some increase in power and size because two PLLs should be included. However, the increase of size is not huge because the R/C filter is replaced with pure digital implementation. And the power consumption can be accepted considering the improvement in jitter performance.

Table. 1.1. Qualitative Compressions between Clock Generator Architectures

Items	Single Voltage CPPLL	Dual Voltage CPPLL	Flying Adder	Hybrid Dual Loop	Single Loop ADPLL	All Digital Dual Loop
Type	Analog		Hybrid		All Digital	
Leakage	X	△	△	△	○	○
Filter size	X	X	△	△	○	○
Jitter[1]	X	△	△	○	X[3]	○[3]
Auto P&R	X	X	△[2]	△[2]	○[4]	○[4]
Size	△	X[5]	X[6]	X[7]	○	△[8]
Power	△	X[9]	X[9]	△[11]	△	△[12]

[1] Assuming the same power consumption and operating voltage.
[1] Assuming that bandwidth is extremely low and ring oscillator VCO and DCO show poor phase noise.
[2] Only digital loop can be synthesized.
[3] Assuming that the quantization is not a limiting factor.
[4] Assuming that TDC and DCO can be implemented with simple logic gates. If not, only digital portion can be synthesized.
[5] Thick gate oxide Transistor is used for loop filter.
[6] Total size is equal to the sum of CPPLL+ ADPLL + Flying Adder.
[7] Total size is equal to the sum of CPPLL+ ADPLL.
[8] Total size is equal to the sum of ADPLL+ ADPLL.
[9] Thick gate oxide transistor block need higher supply voltage.
[10] To generate and drive multi-phase clocks, higher power is required.
[11] Total power is equal to the sum of CPPLL+ ADPLL.
[12] Total power is equal to the sum of ADPLL+ ADPLL. The power consumption of digital block will be reduced in a highly scaled MOS technology.

X = Poor, △ = Fair, and ○ = Excellent

2. Tutorial of ADPLL Design

2.1 Introduction

2.1.1 Motivation for a pure digital

There are many design challenges in conventional charge pump PLL. The all digital PLL (ADPLL) is basically proposed to solve the design issues of charge pump and loop filter block. The conventional charge pump circuit suffers from a limited dynamic range and a up/down current mismatch. And R/C loop filter consumes large size for a narrow loop bandwidth. In addition, the leakage current in a charge pump and a loop filter makes a large fluctuation in the VCO control node and finally degrades a jitter performance. An ADPLL removes these problems by replacing the charge pump and loop filter with a pure digital block.

2.1.2 **Conversion to digital domain**

The basic function of charge pump block provides an electric current for a loop filter; the provided charge amount is proportional to the phase error between two input signals of PFD. To implement the same functionality in digital domain, TDC (Time to Digital Converter) replaces the chare pump and PFD (Phase Frequency Detector), which measures a phase difference and generates an corresponding output digital code. And the TDC output code is entered into the digital loop filter.

A VCO (Voltage Controlled Oscillator) should be converted to a DCO (Digital Controlled Oscillator) because the output of digital loop filter is not an analog voltage but digital one. By converting to a digital domain, the ADPLL solve the design issues like a limited dynamic range and leakage current issue. In addition, the transfer characteristics of TDC and DLF (Digital Loop Filter) are not affected by PVT variation.

2.2 Functional Blocks

In this section, we will suggest basic functional blocks of an ADPLL. To understand an ADPLL better, we are going to overview the conventional CPPLL and compare it with the ADPLL.

2.2.1 **TDC, and PFD/Charge Pump**

TDC coverts a timing difference into a digital code. In the aspect of functionality, the TDC is the same with the sum of a PFD and a charge pump.

Fig. 2.1 illustrates the phase detection blocks for CPPLL and ADPLL respectively. The PFD and charge pump circuit of the conventional CPPLL coverts a phase difference into proportional output charges ΔQ. In a similar way, the TDC of ADPLL generates a digital code which is proportional to the timing difference between FREF and FDIV; D_{OUT} [N:1] is a digital code having N bit word length. While the output DOUT is unit less digital code, this can be considered as an equivalent representation of the ΔQ of CPPLL.

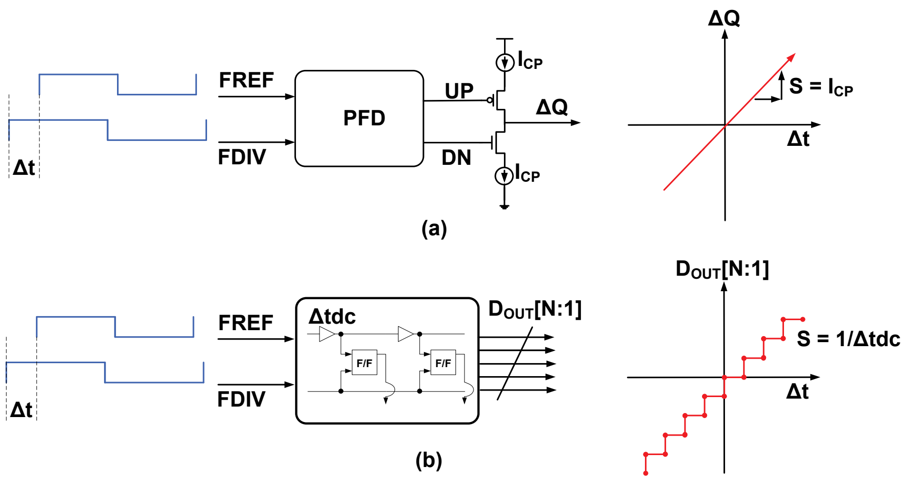


Fig. 2.1. The phase detection blocks of a CPPLL and an ADPLL. (a) the PFD + Charge Pump of a CPPLL. (b) the TDC of an ADPLL

Fig. 2.2 illustrates the detailed operation of phase detection block. There is Up current I_{OUT} because the FREF leads the FDIV. The current flows during the phase difference region Δt, and goes to zero during the other region. The integration of I_{OUT} therefore shows a stair like waveform. That is, the operation of PFD and charge pump shows a non-linear response. To apply s-domain analysis, we should approximate this non-linear system as a linear one. The I_{AVG} of Fig. 2.2 denotes the linear approximation for the non-linear characteristics. We assume that there is an output current I_{AVG} which is constant during a whole period. The amplitude of I_{AVG} is proportional to the phase difference and can be calculated by averaging the integrated I_{OUT} during 1 clock period.

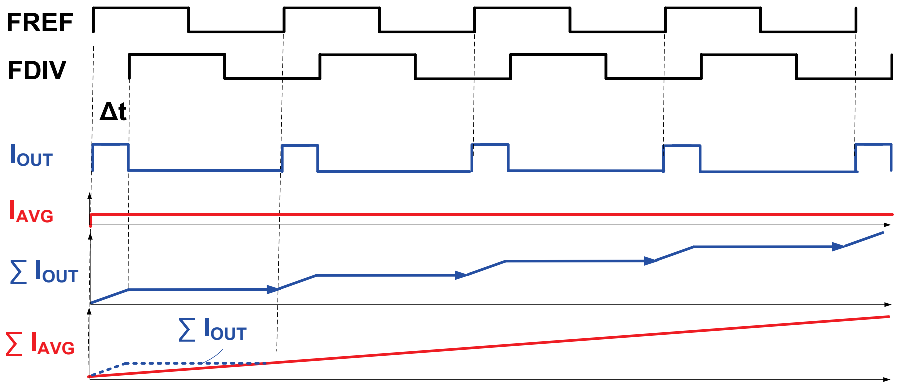


Fig. 2.2. The operation of PFD + Charge pump. And, its linear approximation.

We can express the input timing difference Δt and output average current I_{AVG} in the form of (2.1), where I_{CP} is a chare pump current and T_{REF} is one period of the input reference clock.

$$I_{AVG} = \frac{\Delta t \cdot I_{CP}}{T_{REF}} \; [A] \tag{2.1}$$

To derive the relation between input phase difference and average current output, we should rewrite the time difference Δt as

$$\Delta t = \frac{\Delta\Phi}{2\pi} \cdot T_{REF} \quad [\mathrm{sec}] \tag{2.2}$$

From (2.1) and (2.2), the PFD and charge pump can be described in s-domain as (2.3), where $\Delta\Phi$ is the phase difference of the PFD inputs. The $I_{CP}/2\pi$ is called PFD gain having unit of [A/rad].

$$I_{AVG}(s) = \frac{I_{CP}}{2 \cdot \pi} \cdot \Delta\Phi(s) \ \ [A] \tag{2.3}$$

The TDC gain can be derived through the similar way. The TDC output DOUT is express by (2.4), where $\Delta\tau_{tdc}$ is the unit delay of TDC. Δt and $\Delta\Phi$ are time and phase difference respectively.

$$D_{OUT} = \frac{\Delta t}{\Delta\tau_{tdc}} = \left(\frac{\Delta\Phi}{2\pi} \cdot T_{REF}\right) \cdot \frac{1}{\Delta\tau_{tdc}} = \frac{T_{REF}}{2\pi \cdot \Delta\tau_{tdc}} \cdot \Delta\Phi \ \ [LSB] \tag{2.4}$$

Now, the TDC gain is represented by (2.5).

$$K_{TDC} = \frac{T_{REF}}{2\pi \cdot \Delta\tau_{tdc}} \ \ [1/rad \cdot LSB] \tag{2.5}$$

2.2.2 **Digital Loop Filter and Analog R/C Loop Filter**

In a CPPLL, the current output from a charge pump is entered into a R/C loop filter to move from a current domain to a voltage domain; a capacitor is used to integrate the current. In addition, the loop filter provides a zero to stabilize a PLL loop [23]. The R/C filter of Fig. 2.3 is denoted by (2.6). To derive the relation between R/C filter and digital domain counterpart, bilinear transform is utilized [50]. The s-domain can be rewritten in terms of the z-domain as shown in (2.7). By inserting (2.7) into (2.6). the R/C filter is described in z-domain by (2.8) where T_R denotes the period of operating clock (or sampling clock).

$$H_{LPF}(s) = R + \frac{1}{s \cdot C} \quad [\Omega] \tag{2.6}$$

$$s = \frac{2}{T_R} \cdot \frac{1 - z^{-1}}{1 + z^{-1}} \tag{2.7}$$

$$\begin{aligned} H_{LPF}(z) &= R + \frac{T_R}{2 \cdot C} \cdot \left(\frac{1 + z^{-1}}{1 - z^{-1}} \right) = R + \frac{T_R}{2 \cdot C} \cdot \left(-1 + \frac{2}{1 - z^{-1}} \right) \\ &= \left(R - \frac{T_R}{2 \cdot C} \right) + \frac{T_R / C}{1 - z^{-1}} = \alpha + \frac{\beta}{1 - z^{-1}} \quad [\Omega] \end{aligned} \tag{2.8}$$

From (2.8), the proportional gain α and the integral gain β are expressed as (2.9) and (2.10), where f_R is a sampling frequency.

$$\alpha = R - \frac{T_R}{2 \cdot C} = R - \frac{1}{2 \cdot C \cdot f_R} \tag{2.9}$$

$$\beta = \frac{T_R}{C} = \frac{1}{C \cdot f_R} \tag{2.10}$$

From (2.7), (2.8), (2.9), and (2.10), the trans impedance of (2.8) can be rewritten in s-domain, in terms of α and β as the following.

$$H_{LPF}(s) = R + \frac{1}{s \cdot C} = \left(\alpha + \frac{\beta}{2}\right) + \frac{f_R \cdot \beta}{s}$$

(2.11)

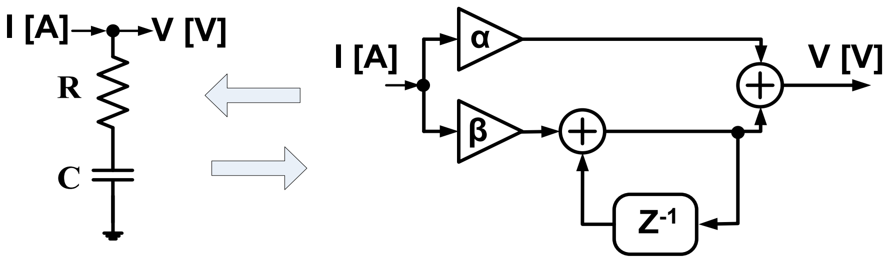


Fig. 2.3. Conversion between analog and digital filter.

Now, we'll show another digital filter used for removing a high frequency noise. It is the 1-st order IIR filter and placed between TDC and the proportional/integral loop filter of Fig. 2.3. The single stage 1st order filter can be cascaded to implement a higher order filter. The loop transfer characteristic is written by a difference equation as shown in (2.12), where **λ** is filter coefficient. The physical meaning of (2.12) is that the filtered output y[k] is the sum of the input x[k] and delayed output y[k-1], and this process averages a high speed fluctuations of y[k]. That is, it performs a low pass filtering.

$$y[k] = (1-\lambda) \cdot y[k-1] + \lambda \cdot x[k]$$

(2.12)

(2.12) is rewritten in z-domain by (2.13).

$$y = (1-\lambda)\cdot z^{-1}\cdot y + \lambda \cdot x \tag{2.13}$$

And it has z-domain loop transfer function of (2.14).

$$H_{IIR}(z) = \frac{\lambda \cdot z}{z-(1-\lambda)} \tag{2.14}$$

From (2.13), we can draw a block diagram of Fig. 2.4.

To get a better physical insight, z-domain loop transfer function is converted to s-domain using bilinear transformation [50]. The z-domain is written in terms of s-domain by (2.15).

$$z = \frac{1+\frac{T_R}{2}\cdot s}{1-\frac{T_R}{2}\cdot s} \tag{2.15}$$

By substituting "z" using (2.15). The equation (2.14) is rewritten by (2.16).

$$H_{IIR}(s) = \frac{1+\frac{s}{2\cdot f_R}}{1+\left(\frac{1}{\lambda}-\frac{1}{2}\right)\cdot\frac{s}{f_R}} \tag{2.16}$$

According to Bodan's work [43], (2.15) and (2.16) can be further approximated assuming $f << f_R$, it means the system operates very slowly compared to the sampling frequency.

$$z = e^{j2\pi f\cdot T_R} \approx 1 + j2\pi f\cdot T_R = 1+\frac{j2\pi f}{f_R} = 1+\frac{s}{f_R} \tag{2.17}$$

To derive (2.18), we utilize (2.17) instead of (2.15).

$$H_{IIR}(s) = \frac{1+\dfrac{s}{f_R}}{1+\dfrac{s}{\lambda f_R}} \tag{2.18}$$

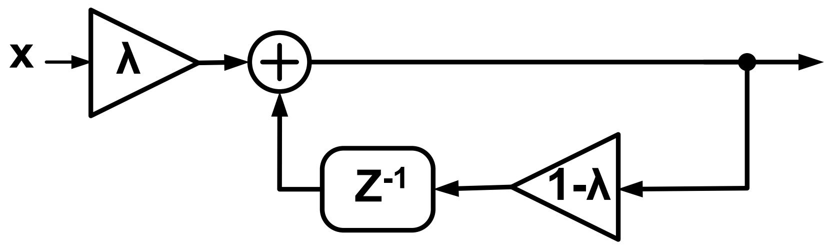


Fig. 2.4. Block diagram of 1st order IIR filter.

The block diagram of Fig. 2.4 is equivalently redrawn in s-domain as Fig. 2.5. It has a low pass transfer function, where ω_{3dB} denotes a 3dB cut-off frequency and has the value of (2.19) [43].

$$\omega_{3dB} = \lambda \cdot f_R \quad [rad/s] \tag{2.19}$$

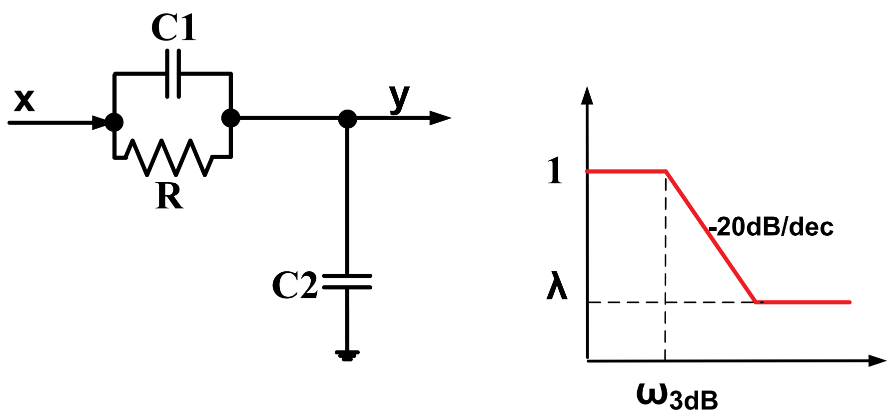


Fig. 2.5. Equivalent R/C filter circuit of Fig. 2.4, and its transfer function.

2.2.3 DCO and VCO

The difference between DCO and VCO is how to control an output frequency. The DCO is controlled in discrete manner by a digital code, whereas VCO is controlled continuously by an analog voltage signal. The output frequency and input control signal of a DCO is determined by (2.20), where Δf_{DCO} denotes a frequency resolution per 1-bit control with the unit of [Hz/LSB] and W is the n-bit control code.

$$f_{DCO} = \Delta f_{DCO} \cdot W[n:1] \ \ [Hz] \tag{2.20}$$

$$f_{VCO} = K_{VCO} \cdot V_{CTRL} \ \ [Hz] \tag{2.21}$$

(2.21) represents the VCO in/out characteristics, where K_{VCO} is the VCO gain with the unit of [Hz/Volt] and V_{CTRL} is an analog control voltage.

The equation of (2.20) and (2.21) should be changed to phase domain to apply for a PLL, because PLL deals with a phase. The conversions are simply completed by multiplying the 2π/S term to execute the integration of angular frequency, which are shown in the next.

$$\Phi_{DCO} = \frac{2\pi}{S} \cdot \Delta f_{DCO} \cdot W[n:1] \ \ [rad] \tag{2.22}$$

$$\Phi_{VCO} = \frac{2\pi}{S} \cdot K_{VCO} \cdot V_{CTRL} \ \ [rad] \tag{2.23}$$

2.2.4 S-domain Model of the Whole Loop

From the results of the previous sections, the conventional CP-PLL can be modeled as illustrated in Fig. 2.6. To simplify an analysis, the loop filter is modeled by 1st order; therefore the whole PLL loop has only 2 poles. We derive an open loop phase transfer function as shown in (2.24). And the closed loop phase transfer function is easily derived as shown in (2.25).

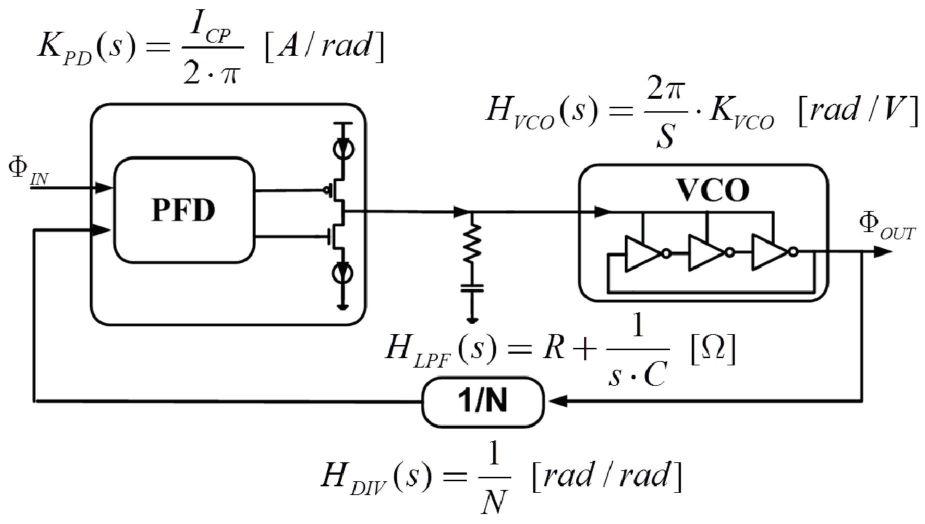


Fig. 2.6. S-domain model of a conventional charge pump PLL.

$$H_{CPPLL}(s) = K_{PD}(s)\times H_{LPF}(s)\times H_{VCO}(s)\times H_{DIV}(s) = \frac{I_{CP}}{2\cdot\pi}\cdot\left(R+\frac{1}{s\cdot C}\right)\cdot\frac{2\pi\cdot K_{VCO}}{S}\cdot\frac{1}{N} \tag{2.24}$$

$$G_{CPPLL}(s) = \frac{\Phi_{OUT}(s)}{\Phi_{IN}(s)} = \frac{N\cdot H_{CPPLL}}{1+H_{CPPLL}} \tag{2.25}$$

In a similar way, we model the ADPLL as illustrated in Fig. 2.7. The open loop phase transfer function is determined by (2.26), and the in/out phase transfer function is shown in (2.27). The loop transfer function H_{LPF} of the DLF (Digital Loop Filter) is converted from the z-domain to s-domain as presented in the section 2.2.2.

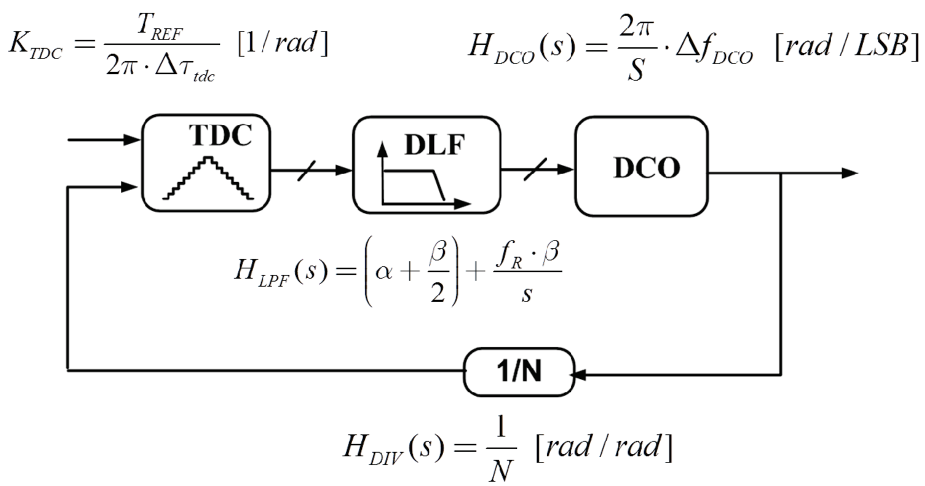


Fig. 2.7. S-domain model of All Digital PLL

$$H_{ADPLL}(s) = K_{TDC}(s) \times H_{LPF}(s) \times H_{DCO}(s) \times H_{DIV}(s)$$
$$= \frac{T_{REF}}{2\pi \cdot \Delta\tau_{tdc}} \cdot \left[\left(\alpha + \frac{\beta}{2}\right) + \frac{f_R \cdot \beta}{s}\right] \cdot \frac{2\pi}{S} \cdot \Delta f_{DCO} \cdot \frac{1}{N} \tag{2.26}$$

$$G_{ADPLL}(s) = \frac{\Phi_{OUT}(s)}{\Phi_{IN}(s)} = \frac{N \cdot H_{ADPLL}}{1 + H_{ADPLL}} \tag{2.27}$$

2.2.5 **ADPLL Loop Design Flow**

CPPLL (Charge Pump PLL) has a well established loop parameter determination procedure [23, 51], thus it will be more convenient to use a CPPLL design flow to determine the loop parameter of an ADPLL. The Kratyuk's work [52] presents a detailed description about ADPLL loop design based on the conventional CPPLL design procedure. Following the design flow, an ADPLL has the proper filter coefficients α and β so that the ADPLL has the same loop transfer characteristics with an equivalent CPPLL. Table. 2.1 summarizes the s-domain models of CPPLL and ADPLL, which are derived in the 2.2.1~ 2.2.4. To make CPPLL and ADPLL have the same response, the both columns of Table. 2.1 should be the same. The column named by "Requirement" show the equivalent relations between CPPLL and ADPLL, and the parts named by "Mapping Relations" is the relation mapping for the CPPLL and ADPLL to have the same transfer characteristics.

Table. 2.1. Summary of S-domain models

	CPPLL, s-domain model	ADPLL, s-domain model
Phase detector	$K_{PFD} = \frac{I_{CP}}{2\pi}\ [A/_{rad}]$	$K_{TDC} = \frac{T_{REF}}{2\pi * \Delta TDC}\ [1/_{rad \cdot LSB}]$
Oscillator	$H_{VCO}(s) = \frac{2\pi \cdot K_{VCO}}{s}\left[\frac{rad}{V}\right]$	$H_{DCO}(s) = \frac{2\pi \cdot \Delta f_{DCO}}{s}\left[\frac{rad}{LSB}\right]$
Loop filter	$H_{LPF}(s) = R + \frac{1}{s \cdot C}$	$H_{DLF}(s) = \left(\alpha + \frac{\beta}{2}\right) + \frac{f_R \cdot \beta}{s}$
Divider	1/N	1/N
Open Loop Gain	$K_{PFD} \times H_{VCO} \times H_{LPF} \times \frac{1}{N}$	$K_{TDC} \times H_{DCO} \times H_{DLF} \times \frac{1}{N}$

Table. 2.2. Loop Parameter Mapping between CPPLL and ADPLL

	Requirements for equivalence	Mapping Relations
Phase detector	$\frac{I_{CP}}{2\pi} = \frac{T_{REF}}{2\pi * \Delta TDC}$	$I_{CP}[A] = \frac{T_{REF}}{\Delta TDC}[{}^{1}/_{LSB}]$
Oscillator	$\frac{2\pi \cdot K_{VCO}}{s} = \frac{2\pi \cdot \Delta f_{DCO}}{s}$	$K_{VCO}[{}^{Hz}/_{V}] = \Delta f_{DCO}[{}^{Hz}/_{LSB}]$
Loop filter	$R + \frac{1}{s \cdot C} = \left(\alpha + \frac{\beta}{2}\right) + \frac{f_R \cdot \beta}{s}$	$\alpha = R - \frac{1}{2 \cdot f_R \cdot C}$ $\beta = \frac{1}{f_R \cdot C}$
Divider	1/N = 1/N	It’s always true.
Open Loop Gain	$H_{CPPLL} = H_{ADPLL}$	It’s true if the prior 4 conditions are met.

We follow the next design procedure to determine the loop parameters of ADPLL.

- Step1 >> Check the design specifications. Find a loop bandwidth f_{BW}, Phase Margin (PM), and multiplying factor (N). In general, loop bandwidth is determined by 1/10 of input reference clock frequency and the phase margin has a value between 50 ~ 70 degree. The multiplication factor N is determined by a target output frequency and input reference clock frequency (N = output frequency / Input frequency).

- Step2 >> Find the resolution of TDC and DCO to meet noise specifications. The TDC resolution ΔTDC and DCO resolution Δf_{DCO} should be determined based on the quantization noise requirements. The procedure to determine ΔTDC and Δf_{DCO} will be covered in the next section.

- Step3 >> In this step, we imagine a CPPLL having the same loop response with the target ADPLL. Find I_{CP} and K_{VCO} by substituting the mapping relations. The I_{CP} and K_{VCO} are defined as the followings.

$$K_{VCO}[{}^{Hz}/_{V}] = \Delta f_{DCO}[{}^{Hz}/_{LSB}]$$

$$I_{CP}[A] = \frac{T_{REF}}{\Delta TDC}[{}^{1}/_{LSB}]$$

- Step4 >> Find the R/C filter values to meet loop bandwidth f_{BW} and Phase Margin (PM) specifications.

$$R = \frac{N}{I_{CP} * K_{vco}[\frac{Hz}{V}]} * \frac{\omega_{BW}^2}{\sqrt{\omega_z^2 + \omega_{BW}^2}} [\Omega]$$

$$\omega_z = \frac{2\pi \cdot f_{BW}}{\tan (PM)}[{}^{rad}/_{s}], \qquad \omega_{BW} = 2\pi \cdot f_{BW}[{}^{rad}/_{s}]$$

$$C = \frac{\tan (PM)}{R * \omega_{BW}} [F]$$

- Step5 >> Find α and β by utilizing the mapping relations of the Table. 2.2.

$$\alpha = R - \frac{1}{2 \cdot f_R \cdot C}$$

$$\beta = \frac{1}{f_R \cdot C}$$

************ This is the end of the procedure. ***********

Although it's helpful for one to follow step 3~5 to understand the whole process of finding α and β. In practice, it's more convenient to directly use the followings [52].

$$\beta = \frac{\Delta TDC * N}{K_{DCO}} * \frac{\omega_{BW}^2}{\sqrt{1 + \tan^2(PM)}} \tag{2.28}$$

$$\alpha = \beta * (\frac{f_{REF} \cdot \tan(PM)}{\omega_{BW}} - \frac{1}{2}) \tag{2.29}$$

2.3 S-domain Noise Model

2.3.1 Noise Transfer Functions

In this part, the transfer functions for individual noise sources are derived. Fig.

2.8 shows the noise model for a charge pump PLL. The red colored circles represent noise sources, where input phase noise Φ_{IN}, charge pump noise Φ_{CP}, loop filter noise Φ_{FILTER}, VCO phase noise Φ_{VCO}, and divider delta-sigma induced noise Φ_{DIV}.

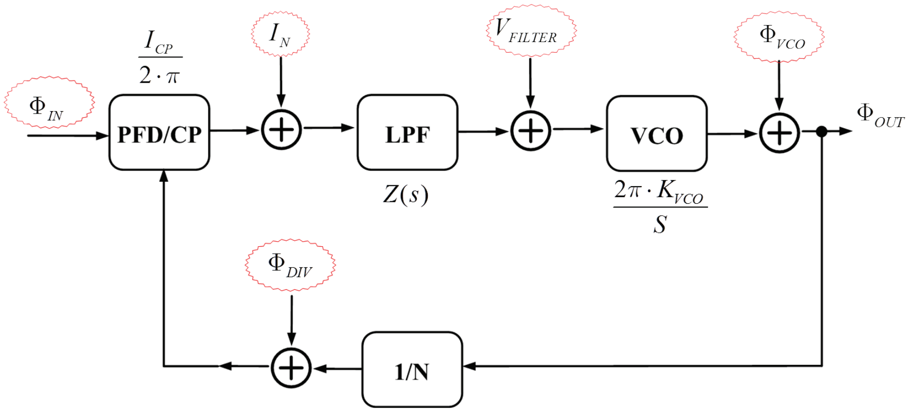


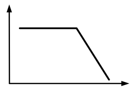
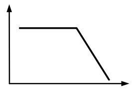
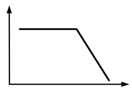
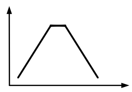
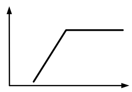
Fig. 2.8. CPPLL Noise Model

We derive the individual noise transfer functions utilizing the same method presented in the section 2.2.4 (page 34). The transfer functions are summarized in Table. 2.3 [23]. H_{open} is the open loop phase transfer function and expressed as (2.30).

$$H_{open}(s) = \frac{I_{CP}}{2\cdot\pi}\cdot Z(s)\cdot\frac{2\pi\cdot K_{VCO}}{S}\cdot\frac{1}{N} \tag{2.30}$$

Fig. 2.9 depicts the noise model of an ADPLL [53], [54]. Noise sources are highlighted by the red colors. From the given block diagram, noise transfer functions can be easily derived. The open loop transfer function is expressed as (2.31), where native DCO resolution Δf_{DCO} is enhanced by the factor of $1/2^{w}$ due to w-bit delta-sigma modulator dithering. The individual noise transfer functions are summarized in Table. 2.4 [54].

Table. 2.3. Noise Transfer Functions of CPPLL

Noise	Transfer Function		
Input clock Noise	$\frac{\Phi_{OUT}}{\Phi_{IN}}$	$N \cdot \frac{H_{open}}{1 + H_{open}}$	
PFD/CP Noise	$\frac{\Phi_{OUT}}{I_{N}}$	$\left(\frac{2\pi}{I_{CP}}\right) \cdot N \cdot \frac{H_{open}}{1 + H_{open}}$	
Divider Noise	$\frac{\Phi_{OUT}}{\Phi_{DIV}}$	$N \cdot \frac{H_{open}}{1 + H_{open}}$	
Filter Noise*	$\frac{\Phi_{OUT}}{V_{FILTER}}$	$\left(\frac{K_{VCO}}{jf}\right) \cdot \frac{1}{1 + H_{open}}$	
VCO Noise	$\frac{\Phi_{OUT}}{\Phi_{VCO}}$	$\frac{1}{1 + H_{open}}$	

$$H_{open} = \frac{T_R}{2\pi \cdot \Delta tdc} \cdot Z(s) \cdot \frac{2\pi \cdot \Delta f_{DCO}}{s \cdot 2^{w}} \cdot \frac{1}{N} \tag{2.31}$$

* $\left(\frac{2\pi \cdot K_{VCO}}{s}\right)$ was simplified to $\left(\frac{K_{VCO}}{jf}\right)$. K_{VCO} has the unit of [Hz/V].

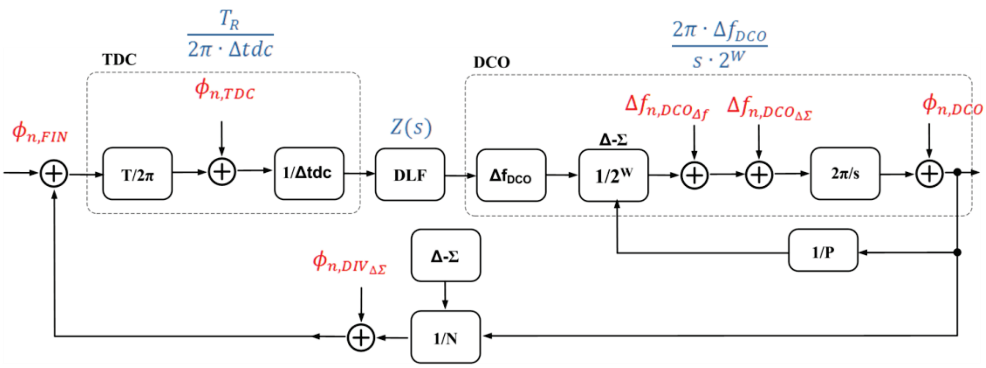


Fig. 2.9. ADPLL Noise Model

By comparing Table. 2.3 and Table. 2.4, we can observe that CPPLL and ADPLL have very similar noise transfer characteristics, which are categorized as the followings.

- Low Pass Transfer

  Input random noise, DSM induced divider noise, TDC, and CP related noises are included in this category.

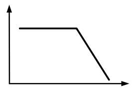
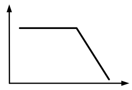
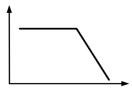
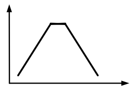
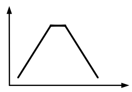
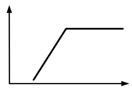
- Band Pass Transfer

  The loop filter noise, DCO dithering, and DCO quantization noise are included in this category. Band pass transfer function has a peak point near the loop bandwidth.

- High Pass Transfer

  DCO and VCO related noises are categorized as this one.

Table. 2.4. Noise Transfer Functions of ADPLL

Noise	Transfer Function		
Input Clock Random Noise	$\frac{\emptyset_{OUT}}{\emptyset_{n,FIN}}$	$N \cdot \frac{H_{open}}{1 + H_{open}}$	
TDC Quantization Noise†	$\frac{\emptyset_{OUT}}{\emptyset_{n,TDC}}$	$\left(\frac{2\pi}{T_R}\right) \cdot N \cdot \frac{H_{open}}{1 + H_{open}}$	
Divider Noise	$\frac{\emptyset_{OUT}}{\emptyset_{n,DIV}}$	$N \cdot \frac{H_{open}}{1 + H_{open}}$	
DCO Dithering Noise‡	$\frac{\emptyset_{OUT}}{\Delta f_{n,DCO_{\Delta\Sigma}}}$	$\frac{1}{jf} \cdot \frac{1}{1 + H_{open}}$	
DCO Quantization Noise§	$\frac{\emptyset_{OUT}}{\Delta f_{n,DCO_{\Delta f}}}$	$\frac{1}{jf} \cdot \frac{1}{1 + H_{open}}$	
DCO Random Noise	$\frac{\emptyset_{OUT}}{\emptyset_{n,DCO}}$	$\frac{1}{1 + H_{open}}$	

To find an output noise due to an input noise source, we should know not only the noise transfer functions (Table. 2.4) but also the input noises. In the following sections, we will present various noise sources.

---

† T_R [sec] is the period of input reference clock.

‡ $\left(\frac{2\pi}{s}\right)$ was simplified to $\left(\frac{1}{jf}\right)$.

§ $\left(\frac{2\pi}{s}\right)$ was simplified to $\left(\frac{1}{jf}\right)$.

2.3.2 **Quantization Noise due to Limited TDC Resolution**

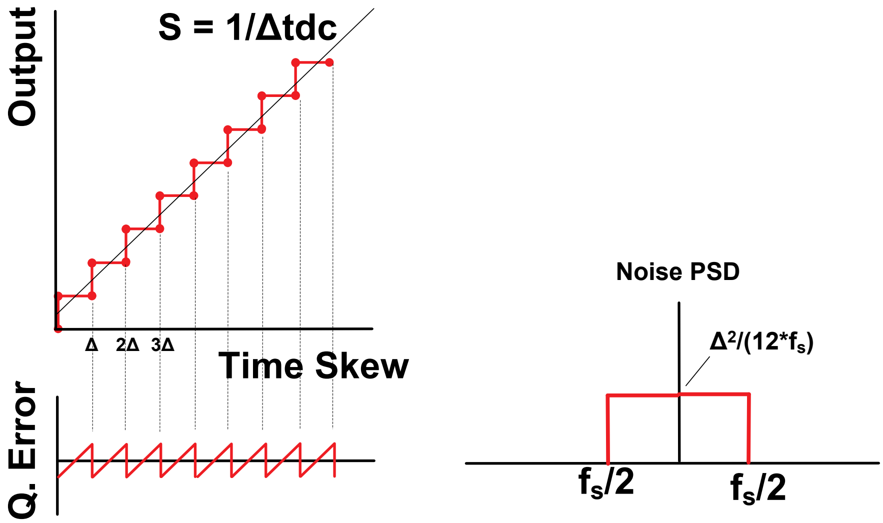


Fig. 2.10. TDC Quantization Noise due to a finite resolution (Δtdc).

Fig. 2.10 shows that a finite TDC resolution generates a quantization noise. Assuming that the noise is uniformly distributed over the nyquist sampling rate, the power spectral density of the quantization error is denoted by (2.32), where $\Delta\mathrm{tdc}$ is TDC resolution and $\mathrm{f_s}$ is the sampling frequency. The sampling frequency is the same with the input reference frequency $\mathrm{f_R}$ in this work.

$$\emptyset_{\mathrm{n,TDC}} = \frac{(\Delta\mathrm{tdc})^2}{12 \cdot \mathrm{f_s}} = \frac{(\Delta\mathrm{tdc})^2}{12 \cdot \mathrm{f_R}} \tag{2.32}$$

The PLL output phase noise due to $\emptyset_{\mathrm{n,TDC}}$ is determined by the convolution with the noise transfer function of Table. 2.4 as the following

$$S_{out,TDC} = \emptyset_{n,TDC} \cdot \left| \left(\frac{2\pi}{T_R}\right) \cdot N \cdot \frac{H_{open}}{1 + H_{open}} \right|^2$$
$$= \frac{(\Delta tdc)^2}{12 \cdot f_R} \cdot \left(\frac{2\pi}{T_R}\right)^2 \cdot \left| N \cdot \frac{H_{open}}{1 + H_{open}} \right|^2 \qquad \left[{}^{rad^2}/_{Hz}\right] \tag{2.33}$$

2.3.3 Quantization Noise due to Divider ΔΣ Noise

The quantization noise due to a delta-sigma modulation is derived in Miller's et.al [55] as shown in (2.34), where m is the order and f_s is the DSM operating frequency, which is the same with the input reference clock frequency in this work.

$$\emptyset_{n,DIV} = \frac{(2\pi)^2}{12 \cdot f_s} \cdot \left(2 \cdot \sin\left(\frac{\pi f}{f_s}\right)\right)^{2(m-1)} \tag{2.34}$$

The PLL output phase noise due to $\emptyset_{n,DIV}$ is expressed by (2.35)

$$S_{out,DIV} = \left\{ \emptyset_{n,DIV} \big|_{f_s = f_R} \right\} \cdot \left| N \cdot \frac{H_{open}}{1 + H_{open}} \right|^2$$
$$= \left\{ \frac{(2\pi)^2}{12 \cdot f_R} \cdot \left(2 \cdot \sin\left(\frac{\pi f}{f_R}\right)\right)^{2(m-1)} \right\} \cdot \left| N \cdot \frac{H_{open}}{1 + H_{open}} \right|^2 \qquad \left[{}^{rad^2}/_{Hz}\right] \tag{2.35}$$

2.3.4 **Quantization Noise due to Limited DCO Resolution**

Like the TDC, DCO has also quantization noise due to its limited frequency resolution Δf_{DCO}. Equation (2.36) shows a frequency quantization error PSD due to a limited frequency resolution, where f_s is a sampling frequency which is the same with an input reference clock without dithering technique. Δf_{DCO} is the DCO frequency resolution [43] [53] [54]. The "sinc" function is added to model the sample and hold operation of a DCO.

$$\Delta f_{n,DCO_{\Delta f}} = \frac{(\Delta f_{DCO})^2}{12 \cdot f_s} \cdot \left\{ \mathrm{sinc}\left(\frac{\Delta f_{DCO}}{f_s}\right)\right\}^2 = \frac{(\Delta f_{DCO})^2}{12 \cdot f_R} \cdot \left\{ \mathrm{sinc}\left(\frac{\Delta f_{DCO}}{f_R}\right)\right\}^2 \left[{}^{Hz^2}/_{Hz} \right] \tag{2.36}$$

In this work, we apply dithering technique to improve the effective frequency resolution. Thus, (2.36) should be rewritten as (2.37), where 'w' is the bit-width of DSM and $f_{\Delta\Sigma}$ denotes the dithering frequency. We can observe that the quantization noise is significantly reduced when $f_{\Delta\Sigma}$ and w is high enough.

$$\Delta f_{n,DCO_{\Delta f}} = \frac{\left(\frac{\Delta f_{DCO}}{2^w}\right)^2}{12 \cdot f_{\Delta\Sigma}} \cdot \left\{ \mathrm{sinc}\left(\frac{f}{f_{\Delta\Sigma}}\right)\right\}^2 \left[{}^{Hz^2}/_{Hz} \right] \tag{2.37}$$

Comparing to (2.36), equation (2.37) show the effective frequency resolution is improved by $\frac{1}{2^w}$, and the sampling frequency is also increased from f_R to $f_{\Delta\Sigma}$. The PLL output noise contribution due to a limited DCO resolution is expressed as

$$S_{out,DCO_{\Delta f}} = \left(\Delta f_{n,DCO_{\Delta f}}\right) \cdot \left|\frac{1}{jf} \cdot \frac{1}{1 + H_{open}}\right|^2$$
$$= \left(\frac{\left(\frac{\Delta f_{DCO}}{2^w}\right)^2}{12 \cdot f_{\Delta\Sigma}} \cdot \left\{\mathrm{sinc}\left(\frac{f}{f_{\Delta\Sigma}}\right)\right\}^2\right) \cdot \left|\frac{1}{jf} \cdot \frac{1}{1 + H_{open}}\right|^2 \left[\frac{rad^2}{Hz}\right] \tag{2.38}$$

In the aspect of frequency error $\Delta f_{n,DCO_{\Delta f}}$, it follows a band pass transfer function $\left|\frac{1}{jf} \cdot \frac{1}{1+H_{open}}\right|$. However, If we define the phase error $\phi_{n,DCO_{\Delta f}}$ as (2.39) then the phase noise due to the limited effective frequency resolution $\frac{\Delta f_{DCO}}{2^w}$ follows a high pass transfer function $\left|\frac{1}{1+H_{open}}\right|$ [54].

$$\phi_{n,DCO_{\Delta f}} = \frac{1}{12} \cdot \left(\frac{\Delta f_{DCO}}{f \cdot 2^w}\right)^2 \cdot \frac{1}{f_{\Delta\Sigma}} \cdot \left\{\mathrm{sinc}\left(\frac{f}{f_{\Delta\Sigma}}\right)\right\}^2 \quad \left[\frac{rad^2}{Hz}\right] \tag{2.39}$$

Now, the equation (2.38) is rewritten as

$$S_{out,DCO_{\Delta f}} = \left(\phi_{n,DCO_{\Delta f}}\right) \cdot \left|\frac{1}{1 + H_{open}}\right|^2$$
$$= \frac{1}{12} \cdot \left(\frac{\Delta f_{DCO}}{f \cdot 2^w}\right)^2 \cdot \frac{1}{f_{\Delta\Sigma}} \cdot \left\{\mathrm{sinc}\left(\frac{f}{f_{\Delta\Sigma}}\right)\right\}^2 \cdot \left|\frac{1}{1 + H_{open}}\right|^2 \left[\frac{rad^2}{Hz}\right] \tag{2.40}$$

2.3.5 **Quantization Noise due to DCO $\Delta\Sigma$ Dithering**

While the dithering technique help improving an effective frequency resolution, it generates additional noise due to a DSM modulation as shown in (2.41), where n is the DSM order. Unlikely the finite resolution effect (2.39), the native frequency resolution Δf_{DCO} is substituted instead of dithered effective frequency resolution $\frac{\Delta f_{DCO}}{2^w}$.

$$\Delta f_{n,DCO_{\Delta\Sigma}} = \frac{(\Delta f_{DCO})^2}{12 \cdot f_{\Delta\Sigma}} \cdot \left\{2 \cdot \sin\left(\frac{\pi f}{f_{\Delta\Sigma}}\right)\right\}^{2n} \ \left[{}^{Hz^2}/_{Hz}\right] \tag{2.41}$$

The PLL output noise contribution due to a DCO dithering is expressed as (2.42).

$$\begin{aligned} S_{out,DCO_{\Delta\Sigma}} &= \left(\Delta f_{n,DCO_{\Delta\Sigma}}\right) \cdot \left|\frac{1}{jf} \cdot \frac{1}{1 + H_{open}}\right|^2 \\ &= \left(\frac{(\Delta f_{DCO})^2}{12 \cdot f_{\Delta\Sigma}} \cdot \left\{2 \cdot \sin\left(\frac{\pi f}{f_{\Delta\Sigma}}\right)\right\}^{2n}\right) \cdot \left|\frac{1}{jf} \cdot \frac{1}{1 + H_{open}}\right|^2 \ \left[\frac{rad^2}{Hz}\right] \end{aligned} \tag{2.42}$$

In the aspect of frequency error $\Delta f_{n,DCO_{\Delta\Sigma}}$, it follows a band pass transfer function $\left|\frac{1}{jf} \cdot \frac{1}{1+H_{open}}\right|$. However, if we re-define the phase error $\phi_{n,DCO_{\Delta\Sigma}}$ as (2.43) then the phase noise due to the limited effective frequency resolution $\frac{\Delta f_{DCO}}{2^w}$ follows a high pass transfer function $\left|\frac{1}{1+H_{open}}\right|$ [54].

$$\phi_{n,DCO_{\Delta\Sigma}} = \frac{1}{12} \cdot \left(\frac{\Delta f_{DCO}}{f}\right)^2 \cdot \frac{1}{f_{\Delta\Sigma}} \cdot \left\{2 \cdot \sin\left(\frac{\pi f}{f_{\Delta\Sigma}}\right)\right\}^{2n} \quad \left[\frac{rad^2}{Hz}\right] \tag{2.43}$$

Now, the equation (2.43) is rewritten as

$$\begin{aligned} S_{out,DCO_{\Delta f}} &= \left(\phi_{n,DCO_{\Delta f}}\right) \cdot \left|\frac{1}{1 + H_{open}}\right|^2 \\ &= \frac{1}{12} \cdot \left(\frac{\Delta f_{DCO}}{f}\right)^2 \cdot \frac{1}{f_{\Delta\Sigma}} \cdot \left\{2 \cdot \sin\left(\frac{\pi f}{f_{\Delta\Sigma}}\right)\right\}^{2n} \cdot \left|\frac{1}{1 + H_{open}}\right|^2 \left[\frac{rad^2}{Hz}\right] \end{aligned} \tag{2.44}$$

2.3.6 **Random Noise of DCO and Input Clock**

Though Hajimiri's model provides good insights to phase noise and jitter of a ring oscillator [3], a transistor level simulation is usually required to find an exact phase noise. We also have utilized a "*Spectre*" simulator to find the phase noise of the ring DCO. The obtained phase noise results ($\phi_{n,DCO}$) are inserted into the noise transfer model shown in Table. 2.4. The input phase noise $\phi_{n,FIN}$ is measured and the result is inserted into the noise transfer function. The PLL output phase noise $S_{out,FIN}$ due to an input clock noise ($\phi_{n,FIN}$) is expressed by (2.45) and it follows a high pass transfer function.

$$S_{out,FIN} = (\phi_{n,FIN}) \cdot \left|\frac{N \cdot H_{open}}{1 + H_{open}}\right|^2 \tag{2.45}$$

The PLL output phase noise $S_{out,DCO}$ due to the DCO random noise ($\phi_{n,DCO}$) is expressed by (2.46) and it follows a high pass transfer function.

$$S_{out,DCO} = (\phi_{n,DCO}) \cdot \left|\frac{1}{1 + H_{open}}\right|^2 \tag{2.46}$$

2.3.7 **Over-all Phase Noise**

The total phase noise S_{out} is calculated by summing the individual noise components of Table. 2.5. The power spectral density of equation (2.47) can be changed to the more widely used phase noise [dBc/Hz], as shown in (2.48). Fig. 2.11 plots the noise transfer functions and individual output noise components of Table. 2.5. Within a loop bandwidth, the TDC quantization and input clock random noise are dominant. Whereas, the DCO related noise and divider DSM noise affect dominantly at an out band region.

$$\mathrm{S_{out,}} = \mathrm{S_{out,FIN}} + \mathrm{S_{out,TDC}} + \mathrm{S_{out,DIV}} + \mathrm{S_{out,DCO_{\Delta\Sigma}}} + \mathrm{S_{out,DCO_{\Delta f}}} + \mathrm{S_{out,DCO}} \tag{2.47}$$

$$\mathcal{L}(\mathrm{f}) = 10 \cdot \log_{10} \mathrm{S_{out,}} \qquad [\frac{\mathrm{dBc}}{\mathrm{Hz}}] \tag{2.48}$$

Table. 2.5. Individual Noise Components of PLL Output.

Noise Source	Symbol	Contribution to PLL output
Input clock	$\mathrm{S_{out,FIN}}$	$(\phi_{\mathrm{n,FIN}}) \cdot \left\lvert \mathrm{N} \cdot \frac{\mathrm{H_{open}}}{1 + \mathrm{H_{open}}} \right\rvert^2$
TDC resolution	$\mathrm{S_{out,TDC}}$	$\frac{(\Delta \mathrm{tdc})^2}{12 \cdot \mathrm{f_R}} \cdot \left(\frac{2\pi}{\mathrm{T_R}}\right)^2 \cdot \left\lvert \mathrm{N} \cdot \frac{\mathrm{H_{open}}}{1 + \mathrm{H_{open}}} \right\rvert^2$
Divider DSM noise	$\mathrm{S_{out,DIV}}$	$\left\{ \frac{(2\pi)^2}{12 \cdot \mathrm{f_R}} \cdot \left(2 \cdot \sin\left(\frac{\pi \mathrm{f}}{\mathrm{f_R}}\right)\right)^{2(\mathrm{m}-1)} \right\} \cdot \left\lvert \mathrm{N} \cdot \frac{\mathrm{H_{open}}}{1 + \mathrm{H_{open}}} \right\rvert^2$
DCO dithering	$\mathrm{S_{out,DCO_{\Delta\Sigma}}}$	$\frac{1}{12} \cdot \left(\frac{\Delta \mathrm{f_{DCO}}}{\mathrm{f}}\right)^2 \cdot \frac{1}{\mathrm{f_{\Delta\Sigma}}} \cdot \left\{2 \cdot \sin\left(\frac{\pi \mathrm{f}}{\mathrm{f_{\Delta\Sigma}}}\right)\right\}^{2\mathrm{n}} \cdot \left\lvert \frac{1}{1 + \mathrm{H_{open}}} \right\rvert^2$
DCO resolution	$\mathrm{S_{out,DCO_{\Delta f}}}$	$\frac{1}{12} \cdot \left(\frac{\Delta \mathrm{f_{DCO}}}{\mathrm{f} \cdot 2^{\mathrm{w}}}\right)^2 \cdot \frac{1}{\mathrm{f_{\Delta\Sigma}}} \cdot \left\{\mathrm{sinc}\left(\frac{\mathrm{f}}{\mathrm{f_{\Delta\Sigma}}}\right)\right\}^2 \cdot \left\lvert \frac{1}{1 + \mathrm{H_{open}}} \right\rvert^2$
DCO random noise	$\mathrm{S_{out,DCO}}$	$(\phi_{\mathrm{n,DCO}}) \cdot \left\lvert \frac{1}{1 + \mathrm{H_{open}}} \right\rvert^2$

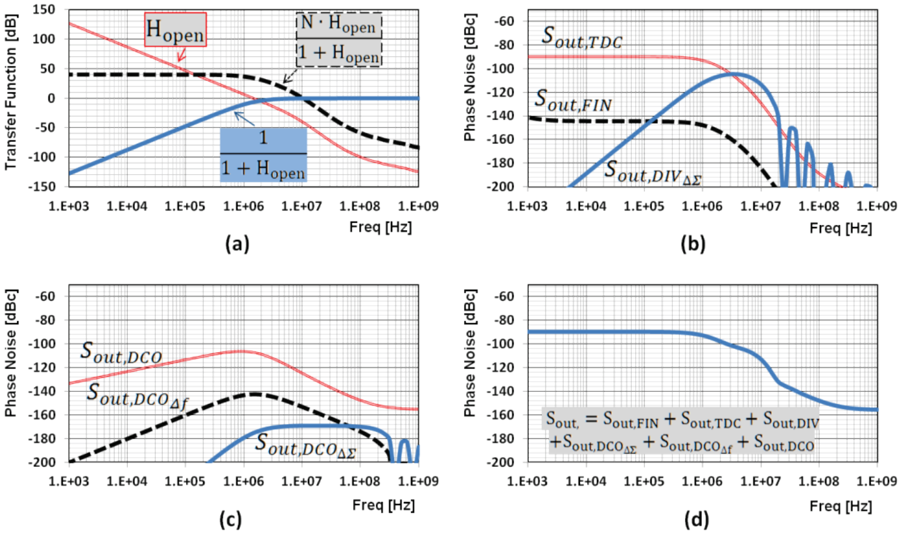


Fig. 2.11. s-domain Noise Modeling. (a) Noise transfer functions, (b) Input referred noise sources, (c) DCO referred noise sources, (d) Over-all noise output.

3. Synthesizable All Digital Pixel Clock PLL Design

3.1 Overview

3.1.1 Introduction of Pixel Clock PLL

Although new display interface standards have been brought into the digital domain, the traditional analog RGB video signal is still widely used. The RGB signal must be converted to the digital domain in order to drive the flat panel digital interface. Fig. 3.1 depicts an Analog Front End (AFE) block diagram for video signal conversion. The Pixel Clock Generator (PCG) regenerates an ADC sampling clock from a very low-frequency reference clock called the horizontal synchronization (HSYNC), which has a frequency range between 10 kHz and 200 kHz [56, 57]. For a good display, the pixel clock must exhibit small integrated jitter and tracking jitter at less than one third of the output clock period [58],[59]. However, an extremely low-frequency reference clock and a limited loop bandwidth make it difficult to significantly reduce the VCO phase noise. While many researchers have proposed jitter reduction techniques for the PCG [28, 46-48, 58-64], there have been few reports of a synthesized PCG suitable for deep submicron processes. We propose a *synthesized all-digital* pixel clock generator that has low integrated jitter, compact size, and low power consumption [65].

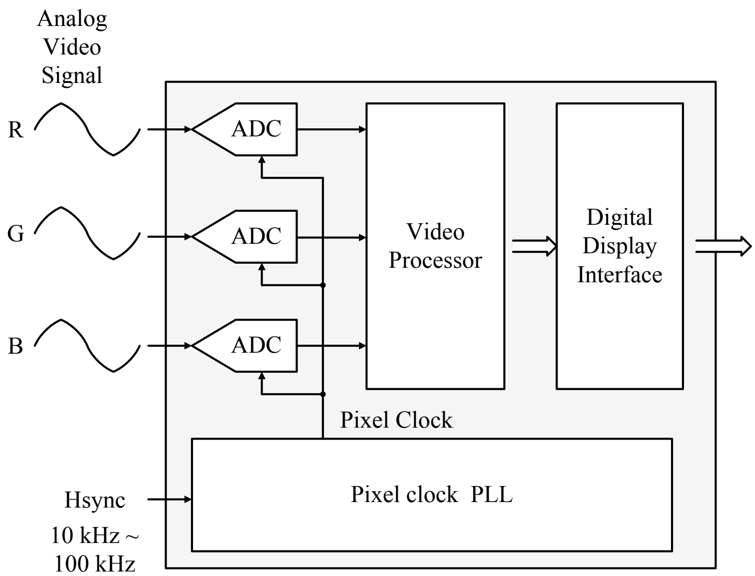


Fig. 3.1. The typical analog front end for flat panel displays.

3.1.1 **Design Specifications**

Table. 3.1, Table. 3.2, Table. 3.3 show the input horizontal synchronization clock (Hsync) and output pixel clock frequency relations for various display standards. Pixel clock generator (PCG) requires a wide range in/out clock frequency ranges. As shown in the tables, the input clock ranges from 10 kHz to 200 kHz and output pixel clock should cover from 13.5MHz to 550MHz.

PCG should meet stringent jitter specification for a good display quality. But the extremely low input frequency limits a PLL loop bandwidth, thus we cannot filter out the VCO (or DCO) noise significantly. In conclusion, the DCO must have good phase noise performance to meet stringent jitter specification under low loop bandwidth condition. The design specification of PCG is summarized as the followings.

- Extremely low input frequency : 10 kHz ~ 200 kHz
- Wide output frequency ranges : 13.5 MHz ~ 550 MHz
- Low integrated jitter: Less than 10% ~30% of pixel clock period.

To implement the design requirements in the ADPLL, we must meet additional design specifications listed in the followings.

- Both Fine TDC resolution and wide detection range to cover low input frequency and large multiplication factor
- Both Fine DCO resolution and wide tuning range to cover large multiplication factor and wide output frequency range
- Low DCO phase noise to meet over-all jitter specification under low

PLL loop bandwidth.

To achieve the listed design requirements, we should solve some challenges in TDC and DCO design.

- Larger hardware size to achieve both fine resolution and wide range.
- Ring oscillator DCO is required to meet wide tuning range, but it has poor phase noise.

In this work, we'll propose a new all digital PLL architecture to meet design requirements. Additionally, to improve the design efficiency, whole design is described in HDL and automatically synthesized using commercially available P&R tool.

Table. 3.1. Video Standard Formats (Captured from [56])

Description (Mode)	Active Pixels	Active Lines	P/I	Vsync	Hsync (kHz)	Pixel Clk (MHz)	Total Pixel	Total Line
480i	704	480	I	59.940	15.734	13.500	858	262.5
480i	704	480	I	60.000	15.750	13.514	858	262.5
480p	704	480	P	60.000	31.500	27.027	858	525
480p	704	480	P	59.940	31.469	27.000	858	525
576i	720	576	I	50.000	15.625	13.500	864	312.5
576p	720	576	P	50.000	31.250	27.000	864	625
720p	1280	720	P	50.000	37.500	74.250	1980	750
720p	1280	720	P	59.940	44.955	74.176	1650	750
720p	1280	720	P	60.000	45.000	74.250	1650	750
1080i	1920	1080	I	59.940	33.716	74.176	2200	562.5
1080i	1920	1080	I	50.000	28.125	74.250	2640	562.5
1080i	1920	1080	I	60.000	33.750	74.250	2200	562.5
VGA	640	480	P	59.940	31.469	25.175	800	525
VGA	640	480	P	72.809	37.861	31.500	832	520
VGA	640	480	P	75.000	37.500	31.500	840	500
VGA	640	480	P	85.008	43.269	36.000	832	509
VGA_B	720	400	P	85.039	37.927	35.500	936	446
VGAi	640	480	I	59.940	15.734	12.588	800	262.5
VGAi	640	480	I	60.000	15.750	12.600	800	262.5
SVGA	800	600	P	56.250	35.156	36.000	1024	625
SVGA	800	600	P	60.317	37.879	40.000	1056	628
SVGA	800	600	P	72.188	48.077	50.000	1040	666
SVGA	800	600	P	75.000	46.875	49.500	1056	625
SVGA	800	600	P	85.061	53.674	56.250	1048	631
XGAi	1024	768	I	86.958	35.522	44.900	1264	408.5
XGA	1024	768	P	60.004	48.363	65.000	1344	806
XGA	1024	768	P	70.069	56.476	75.000	1328	806
XGA	1024	768	P	75.029	60.023	78.750	1312	800
XGA	1024	768	P	84.997	68.677	94.500	1376	808
XGA_B	1152	864	P	75.000	67.500	108.000	1600	900
SXGA	1280	1024	P	60.020	63.981	108.000	1688	1066
WXGA	1280	960	P	60.000	60.000	108.000	1800	1000
WXGA	1280	768	P	59.995	47.396	68.250	1440	790
WXGA	1280	768	P	59.870	47.776	79.500	1664	798
WXGA	1280	768	P	74.893	60.289	102.250	1696	805
WSXGA	1400	1050	P	59.948	64.744	101.000	1560	1080
WSXGA	1366	768	P	60.000	47.700	85.500	1792	795
WSXGA	1360	768	P	60.015	47.712	85.500	1792	795

Table. 3.2. Video Standard Formats (Captured from [57])

Pixel Format	Refresh Rate	Horizontal Frequency	Pixel Frequency	Standard Type	Original Document	Date
640 x 350	85 Hz	37.9 kHz	31.500 MHz	VESA Standard	VDMTPROP	3/1/96
640 x 400	85 Hz	37.9 kHz	31.500 MHz	VESA Standard	VDMTPROP	3/1/96
720 x 400	85 Hz	37.9 kHz	35.500 MHz	VESA Standard	VDMTPROP	3/1/96
640 x 480	60 Hz	31.5 kHz	25.175 MHz	Industry Standard	n/a	n/a
	72 Hz	37.9 kHz	31.500 MHz	VESA Standard	VS901101	12/2/92
	75 Hz	37.5 kHz	31.500 MHz	VESA Standard	VDMT75HZ	10/4/93
	85 Hz	43.3 kHz	36.000 MHz	VESA Standard	VDMTPROP	3/1/96
800 x 600	56 Hz	35.2 kHz	36.000 MHz	VESA Guidelines	VG900601	8/6/90
	60 Hz	37.9 kHz	40.000 MHz	VESA Guidelines	VG900602	8/6/90
	72 Hz	48.1 kHz	50.000 MHz	VESA Standard	VS900603A	8/6/90
	75 Hz	46.9 kHz	49.500 MHz	VESA Standard	VDMT75HZ	10/4/93
	85 Hz	53.7 kHz	56.250 MHz	VESA Standard	VDMTPROP	3/1/96
	120 Hz (RB)	76.3 kHz	73.250 MHz	CVT Red. Blanking	n/a	5/1/07
848 x 480	60 Hz	31.0 kHz	33.750 MHz	VESA Standard	AddDMT	3/4/03
1024 x 768	43 Hz (Int.)	35.5 kHz	44.900 MHz	Industry Standard	n/a	n/a
	60 Hz	48.4 kHz	65.000 MHz	VESA Guidelines	VG901101A	9/10/91
	70 Hz	56.5 kHz	75.000 MHz	VESA Standard	VS910801-2	8/9/91
	75 Hz	60.0 kHz	78.750 MHz	VESA Standard	VDMT75HZ	10/4/93
	85 Hz	68.7 kHz	94.500 MHz	VESA Standard	VDMTPROP	3/1/96
	120 Hz (RB)	97.6 kHz	115.500 MHz	CVT Red. Blanking	n/a	5/1/07
1152 x 864	75 Hz	67.5 kHz	108.000 MHz	VESA Standard	VDMTPROP	3/1/96
1280 x 768	60 Hz(RB)	47.4 kHz	68.250 MHz	CVT Red. Blanking	AddDMT	3/4/03
	60 Hz	47.8 kHz	79.500 MHz	CVT	AddDMT	3/4/03
	75 Hz	60.3 kHz	102.250 MHz	CVT	AddDMT	3/4/03
	85 Hz	68.6 kHz	117.500 MHz	CVT	AddDMT	3/4/03
	120 Hz (RB)	97.4 kHz	140.250 MHz	CVT Red. Blanking	n/a	5/1/07
1280 x 800	60 Hz(RB)	49.3 kHz	71.000 MHz	CVT Red. Blanking	CVT1.02MA-R	5/1/07
	60 Hz	49.7 kHz	83.500 MHz	CVT	CVT 1.02MA	5/1/07
	75 Hz	62.8 kHz	106.500 MHz	CVT	CVT 1.02MA	5/1/07
	85 Hz	71.6 kHz	122.500 MHz	CVT	CVT 1.02MA	5/1/07
	120 Hz (RB)	101.6 kHz	146.250 MHz	CVT Red. Blanking	n/a	5/1/07
1280 x 960	60 Hz	60.0 kHz	108.000 MHz	VESA Standard	VDMTPROP	3/1/96
	85 Hz	85.9 kHz	148.500 MHz	VESA Standard	VDMTPROP	3/1/96
	120 Hz (RB)	121.9 kHz	175.500 MHz	CVT Red. Blanking	n/a	5/1/07

Table. 3.3. Video Standard Formats (Captured from [57])

Pixel Format	Refresh Rate	Horizontal Frequency	Pixel Frequency	Standard Type	Original Document	Date
1280 x 1024	60 Hz	64.0 kHz	108.000 MHz	VESA Standard	VDMTREV	12/18/96
	75 Hz	80.0 kHz	135.000 MHz	VESA Standard	VDMT75HZ	10/4/93
	85 Hz	91.1 kHz	157.500 MHz	VESA Standard	VDMTPROP	3/1/96
	120 Hz (RB)	130.0 kHz	187.250 MHz	CVT Red. Blanking	n/a	5/1/07
1360 x 768	60 Hz	47.7 kHz	85.500 MHz	VESA Standard	AddDMT	3/4/03
	120 Hz (RB)	97.5 kHz	148.250 MHz	CVT Red. Blanking	n/a	5/1/07
1400 x 1050	60 Hz(RB)	64.7 kHz	101.000 MHz	CVT Red. Blanking	AddDMT	5/13/03
	60 Hz	65.3 kHz	121.750 MHz	CVT	AddDMT	3/4/03
	75 Hz	82.3 kHz	156.000 MHz	CVT	AddDMT	3/4/03
	85 Hz	93.9 kHz	179.500 MHz	CVT	AddDMT	3/4/03
	120 Hz (RB)	133.3 kHz	208.000 MHz	CVT Red. Blanking	n/a	5/1/07
1440 x 900	60 Hz(RB)	55.5 kHz	88.750 MHz	CVT Red. Blanking	CVT1.30MA-R	7/14/04
	60 Hz	55.9 kHz	106.500 MHz	CVT	CVT 1.30MA	7/14/04
	75 Hz	70.6 kHz	136.750 MHz	CVT	CVT 1.30MA	7/14/04
	85 Hz	80.4 kHz	157.000 MHz	CVT	CVT 1.30MA	7/14/04
	120 Hz (RB)	114.2 kHz	182.750 MHz	CVT Red. Blanking	n/a	5/1/07
1600 x 1200	60 Hz	75.0 kHz	162.000 MHz	VESA Standard	VDMTREV	12/18/96
	65 Hz	81.3 kHz	175.500 MHz	VESA Standard	VDMTREV	12/18/96
	70 Hz	87.5 kHz	189.000 MHz	VESA Standard	VDMTREV	12/18/96
	75 Hz	93.8 kHz	202.500 MHz	VESA Standard	VDMTREV	12/18/96
	85 Hz	106.3 kHz	229.500 MHz	VESA Standard	VDMTREV	12/18/96
	120 Hz (RB)	152.4 kHz	268.250 MHz	CVT Red. Blanking	n/a	5/1/07
1680 x 1050	60 Hz(RB)	64.7 kHz	119.000 MHz	CVT Red. Blanking	CVT1.76MA-R	7/14/04
	60 Hz	65.3 kHz	146.250 MHz	CVT	CVT 1.76MA	7/14/04
	75 Hz	82.3 kHz	187.000 MHz	CVT	CVT 1.76MA	7/14/04
	85 Hz	93.9 kHz	214.750 MHz	CVT	CVT 1.76MA	7/14/04
	120 Hz (RB)	133.4 kHz	245.500 MHz	CVT Red. Blanking	n/a	5/1/07
1792 x 1344	60 Hz	83.6 kHz	204.750 MHz	VESA Standard	VDMTREV	9/17/98
	75 Hz	106.3 kHz	261.000 MHz	VESA Standard	VDMTREV	9/17/98
	120 Hz (RB)	170.7 kHz	333.250 MHz	CVT Red. Blanking	n/a	5/1/07
1856 x 1392	60 Hz	86.3 kHz	218.250 MHz	VESA Standard	VDMTREV	9/17/98
	75 Hz	112.5 kHz	288.000 MHz	VESA Standard	VDMTREV	9/17/98
	120 Hz (RB)	176.8 kHz	356.500 MHz	CVT Red. Blanking	n/a	5/1/07
1920 x 1200	60 Hz(RB)	74.0 kHz	154.000 MHz	CVT Red. Blanking	AddDMT	3/4/03
	60 Hz	74.6 kHz	193.250 MHz	CVT	AddDMT	3/4/03
	75 Hz	94.0 kHz	245.250 MHz	CVT	AddDMT	3/4/03
	85 Hz	107.2 kHz	281.250 MHz	CVT	AddDMT	3/4/03
	120 Hz (RB)	152.4 kHz	317.000 MHz	CVT Red. Blanking	n/a	5/1/07
1920 x 1440	60 Hz	90.0 kHz	234.000 MHz	VESA Standard	VDMTREV	9/17/98
	75 Hz	112.5 kHz	297.000 MHz	VESA Standard	VDMTREV	9/17/98
	120 Hz (RB)	182.9 kHz	380.500 MHz	CVT Red. Blanking	n/a	5/1/07
2560 x 1600	60 Hz (RB)	98.7 kHz	268.500 MHz	CVT Red. Blanking	CVT4.10MA-R	5/1/07
	60 Hz	99.5 kHz	348.500 MHz	CVT	CVT 4.10MA	5/1/07
	75 Hz	125.4 kHz	443.250 MHz	CVT	CVT 4.10MA	5/1/07
	85 Hz	142.9 kHz	505.250 MHz	CVT	CVT 4.10MA	5/1/07
	120 Hz (RB)	203.2 kHz	552.750 MHz	CVT Red. Blanking	n/a	5/1/07

3.2 Proposed Architecture

3.2.1 All Digital Dual Loop PLL

Fig. 3.2 shows the block diagram of an all-digital dual-loop pixel clock generator, which was briefly introduced in our previous work [65]. The DCO is implemented with an all-digital fractional-N PLL to improve the phase noise performance. The phase noise of the fast-loop PLL is superior to that of a conventional ring oscillator-based DCO because the DCO noise is filtered by the fast-loop PLL that has a wide loop bandwidth and a clean input clock from an external crystal oscillator. The output frequency of the fast-loop is determined by (3.1).

$$F_{OUT} = F_{IN} \times M/(S \times 2) \tag{3.1}$$

F_{IN} is the frequency of the external crystal oscillator while M is the divide ratio of the feedback divider in the fast-loop that is controlled by the slow loop. The output frequency of the fast-loop depends only on the divide values M and S regardless of PVT variations. The post *S-divider* is utilized to extend the tuning ranges of the DCO. The IIR filter of the fast-loop is added to reject the delta-sigma noise, and it operates at a higher frequency than the input clock (F_{IN}) in order to reduce the loop latency. The operating frequency of the IIR filter is set by dividing the DCO frequency (F_{DCO}) as shown in Fig. 3.2.

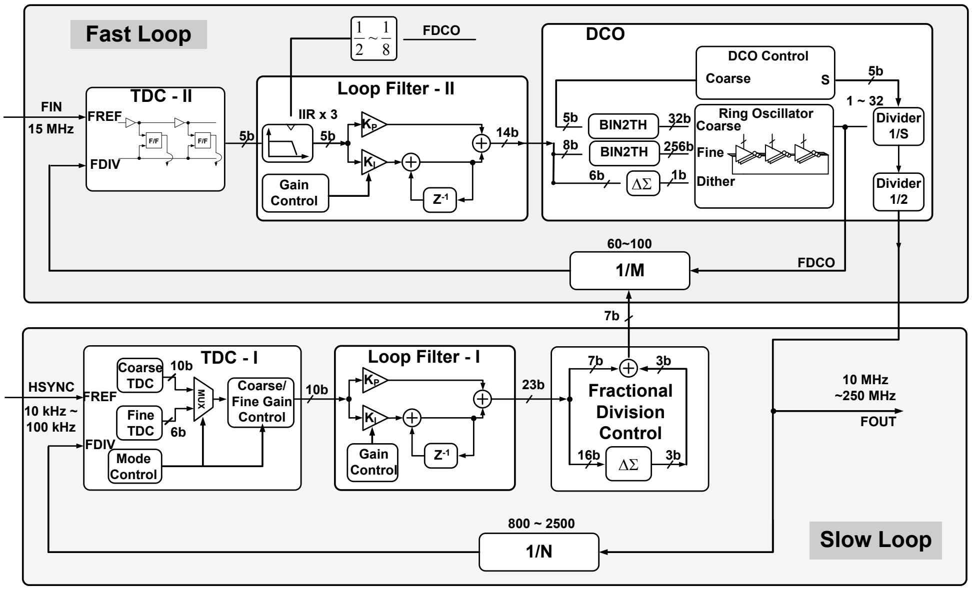


Fig. 3.2. The schematic of the proposed all-digital dual-loop architecture.

3.2.2 2-step controlled TDC

The TDC resolution (Δtdc) needed to meet a target in-band phase noise ($\mathcal{L}$) is expressed by (3.2), and the calculation results are presented in Table. 3.4. Δtdc depends on the reference frequency (f_r), in-band phase noise specification ($\mathcal{L}$) and multiplying factor (*N*) [66].

$$\Delta tdc = \sqrt{\left(10^{(\mathcal{L}/_{10})}\right) \times \left(\frac{12}{(2\pi)^2}\right) \times \frac{1}{N^2} \times \frac{1}{f_r}} \tag{3.2}$$

The slow-loop TDC (TDC-I) should have a large detection range that reduces the excessive lock time as there is a large phase error during the locking process from the extremely low input frequency (*HSYNC*, 10 kHz-200 kHz) and large multiplying factor (*N*, 800~2500).

Table. 3.4. TDC Resolution for Each Loop

Loop	Parameters	Min/Max Ranges	Max TDC Resolution
Fast Loop	Target Phase noise ($\mathcal{L}$)	-95 dBc/Hz	25 ps ~ 42 ps
	Dividing Ratio (M)	60~100	
	Reference Frequency (fr=F_{IN})	15 MHz	
Slow Loop	Target Phase noise ($\mathcal{L}$)	-95 dBc/Hz	12 ps~120 ps
	Dividing Ratio (N)	800 ~ 2500	
	Reference Frequency (fr=HSYNC)	10 kHz ~ 200 kHz	

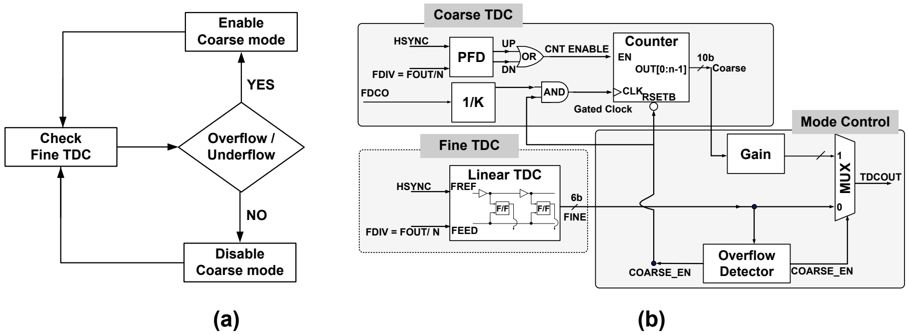


Fig. 3.3. TDC employing 2-step detection architecture. (a) Mode control algorithm and (b) its schematic diagram.

In order to meet the requirements of both the wide detection range and fine resolution, the TDC-I is implemented with a 2-step architecture that consists of coarse and fine TDCs as shown in Fig. 3.3. The coarse mode utilizes a 10-bit frequency counter for the wide detection range, and the fine-TDC is implemented with the conventional delay-cell based linear TDC (64 stages) for the fine resolution [67] as shown in

Fig. 3.4. The relation between input timing skew and TDC output is depicted in Fig. 3.5. More detailed description of the TDC operation can be found in [68].

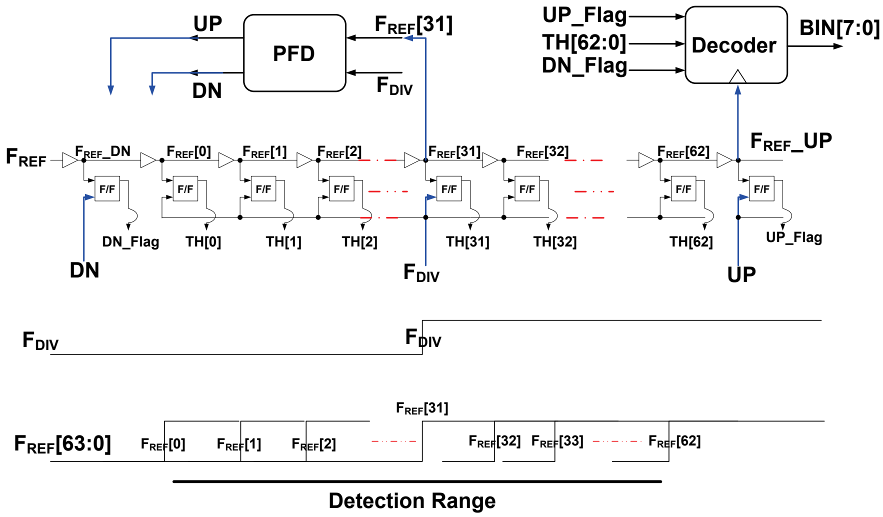


Fig. 3.4. Linear TDC for fine mode.

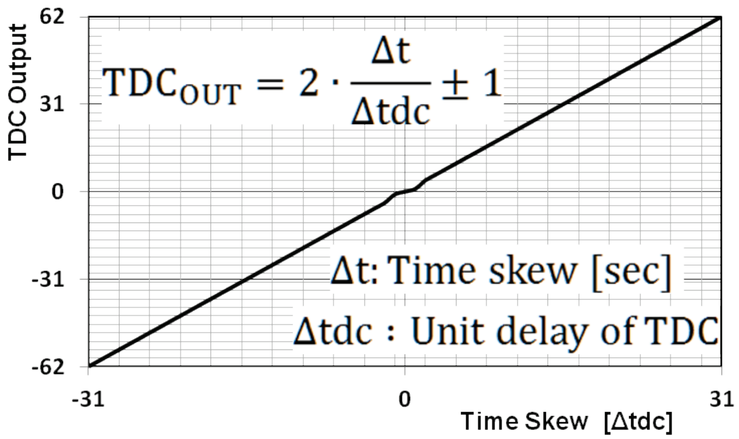


Fig. 3.5. TDC response.

The mode change is controlled by monitoring the status of the fine TDC as shown in Fig. 3.3 (a). When the fine TDC is within its min/max range, the TDC operates in the fine mode while the coarse mode is disabled to save power. The coarse TDC is only enabled when the output of the fine TDC reaches its min/max values. Unlike the prior-art gated ring oscillator TDC [69], there is no additional cost in implementing the coarse TDC as the ring oscillator DCO is reused to drive the counter of the coarse TDC. The resolution and detection range of the coarse TDC are controlled by the frequency divider to deal with the wide range of the input clock signal and various divide ratios. Unlike the slow loop, the TDC of the fast-loop has a narrow detection range, because the input clock frequency is fixed and is relatively high (15 MHz). Therefore, the fast-loop utilizes the conventional linear TDC composed of 32 unit delay stages programmable with time delay.

3.2.3 **3-step controlled DCO**

Fig. 3.6 presents the proposed DCO having 3-step control logic and ring oscillator core. it shows that the DCO is composed of a ring oscillator and control logic gates. The fine control and dithering input of the ring oscillator come from the output of the loop filter (DLF-II). But the coarse and S-values does not come from the filter but is determined from the state of the fine code. Fig. 3.7 show the detailed block diagram of the coarse and S-value control logic. And Fig. 3.8 shows the ring oscillator DCO core having coarse and fine controls. The *coarse* control cell is implemented with a conventional tri-state inverter and the frequency resolution is determined by the minimum available transistor size. However, the minimum device size limits the achievable fine resolution. It would be possible to enhance the resolution by increasing the gate length of the transistors to reduce the on-state current. But this approach increases the gate capacitance and thereby slows down the oscillation frequency.

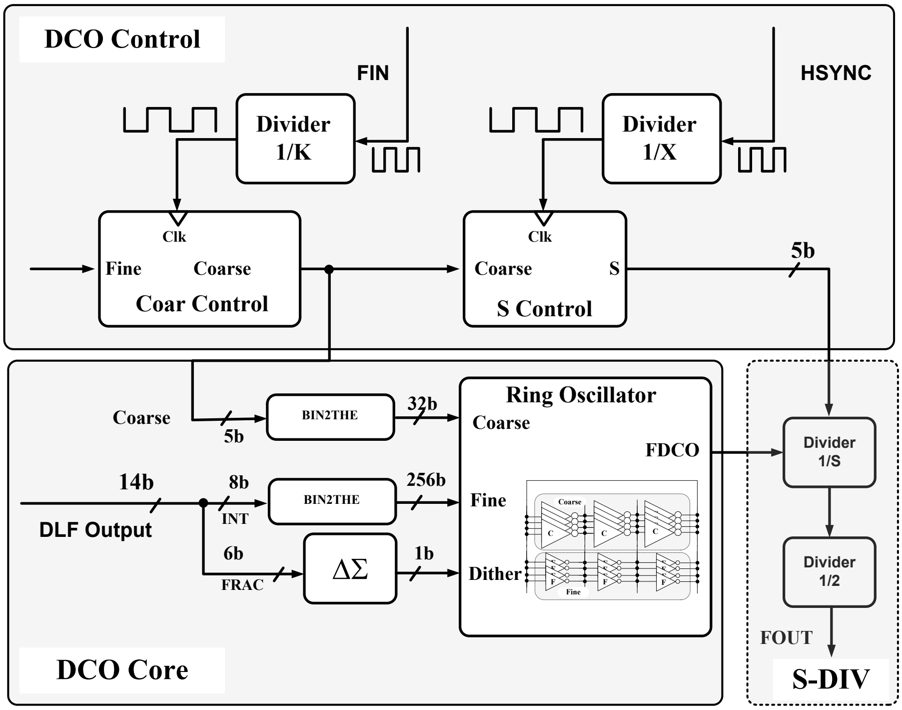


Fig. 3.6. The proposed DCO having 3-step control logic and ring oscillator core.

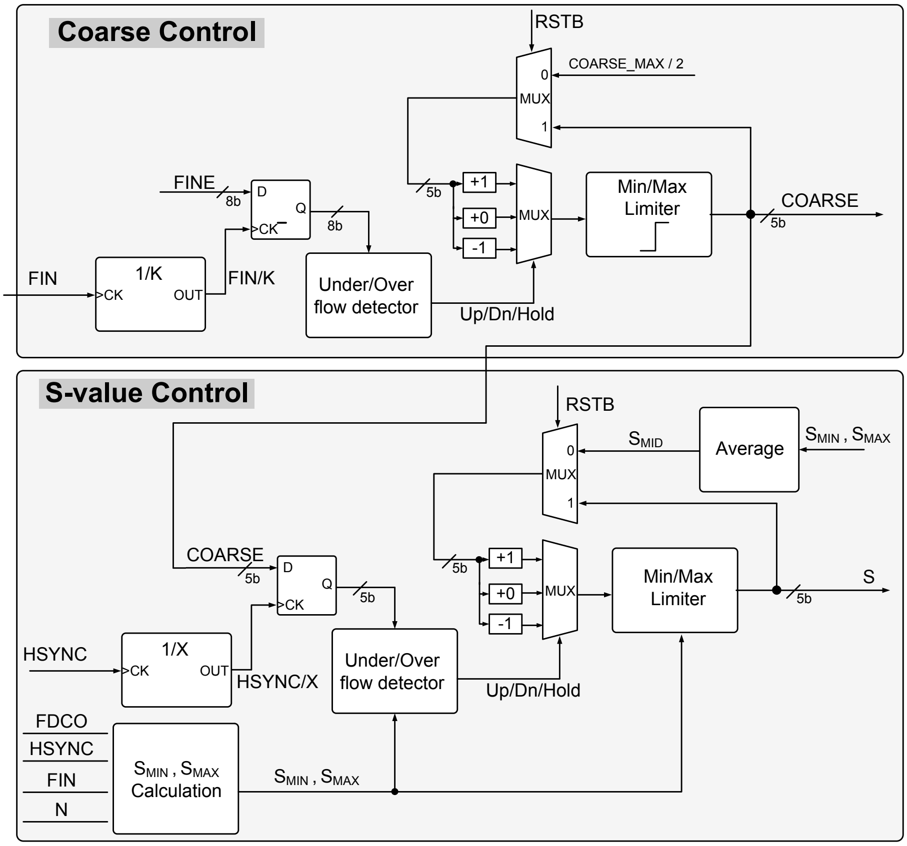


Fig. 3.7. Coarse and S-value Control block

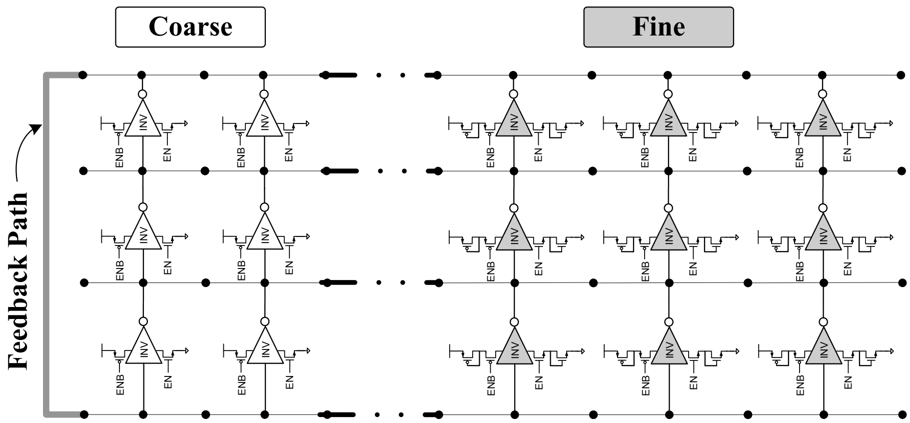


Fig. 3.8. Synthesizable 3-stage ring oscillator DCO. The fine delay cell has diode connected loads to reduce the current difference between on/off states.

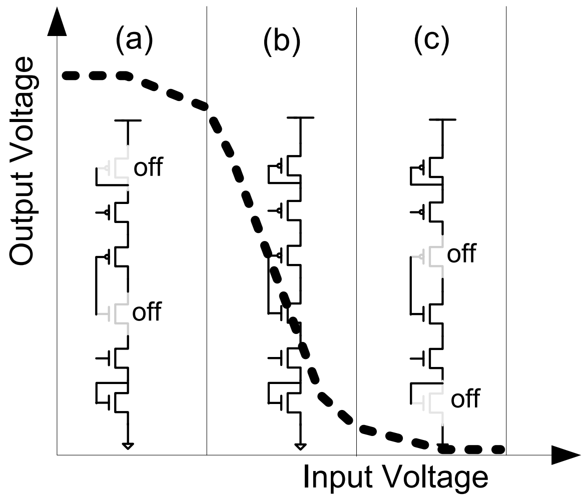


Fig. 3.9. The input/output transfer curve of diode connected fine delay cell. In regions (a) and (c), the fine delay cell is turned off because coarse delay cells force the oscillation node to swing rail to rail.

The proposed delay cell used for the fine control is shown in Fig. 3.9 [65]. The diode connected load reduces the effective drain-source voltage for the tri-state inverter cell and enhances the frequency resolution by reducing the on/off current difference, which is shown in Fig. 3.9 with the operation of the proposed delay cell. The voltage swing is not limited by the diode connected load but determined by the *coarse* delay cell since the *fine* tuning cell is automatically turned off as the voltage swing approaches the power/ground level. The fine control block is implemented with 256 unit delay cells and has a resolution of 1 MHz/LSB. The effective resolution is further enhanced to 15 kHz utilizing 6-bit DSM dithering. The coarse control gain is 40 MHz/LSB composed of 32 delay cells.

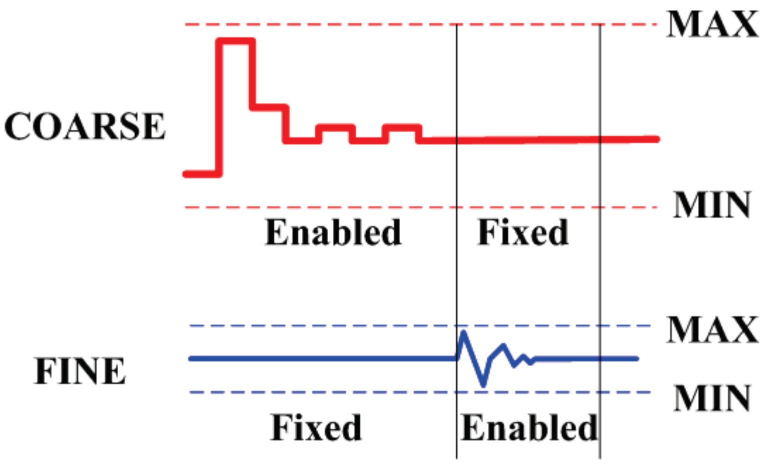


Fig. 3.10. Conventional top-down DCO control

A conventional multi-step controlled DCO (
Fig. 3.10) generally starts up at the coarse mode (fixing the fine code) and moves onto the fine control mode (after fixing the coarse code). This coarse/fine 2-step control scheme has been widely used for the DCO to meet the wide tuning range and fine resolution [66]. However, this kind of top-down sequence cannot be used for the current *dual-loop* architecture, because the feedback divider value M is not fixed until the entire dual loop settles down. The *coarse* and *fine* codes cannot be fixed since the output frequency is determined by the M and S values as shown in equation (3.1). The M and S values should be fixed to utilize the conventional top-down control method, yet those values cannot be fixed unless the *coarse* and *fine* codes are fixed. From this chicken or egg problem, there must be iterations between the coarse and fine controls until the PLL reaches the locked state. Eventually this fundamental issue increases the lock time and disturbs the locking process (Fig. 3.11). Unlike the pure digital dual-loop architecture, Lee's hybrid PLL

(Fig. 1.15) [63] does not suffer from this problem, because the analog VCO is used for the fast-loop and does not have the coarse/fine control.

In order to reduce the iterations between *coarse* and *fine* modes, this work utilizes the bottom-up DCO control algorithm [65]. Fig. 3.12 shows that the *coarse* code is updated whenever the *fine* code reaches its min/max values. When the *fine* code reaches an upper limit, the *coarse* code increases incrementally by 1, while if the *fine* code reaches a lower limit, the *coarse* code decreases by 1. Thus, the *coarse* code is adjusted so that the *fine* code is within the limited region. In the control hierarchy, the *S* divider control is in the upper level of coarse tuning.

Fig. 3.13 illustrates the *S* value that is updated whenever the *coarse* code exceeds its min/max limits. If the *coarse* code reaches the maximum value, then *S* decreases by 1 to reduce the DCO frequency, and vice versa. When the *coarse* code is within the min/max range, *S* holds the current value.

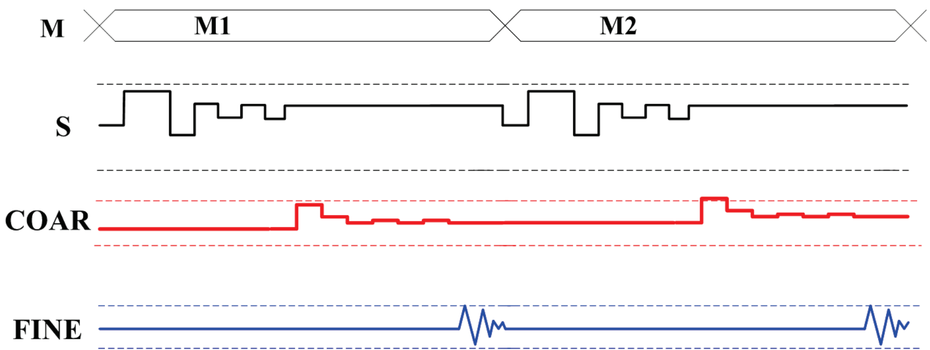


Fig. 3.11. Lock time issue in top-down method.

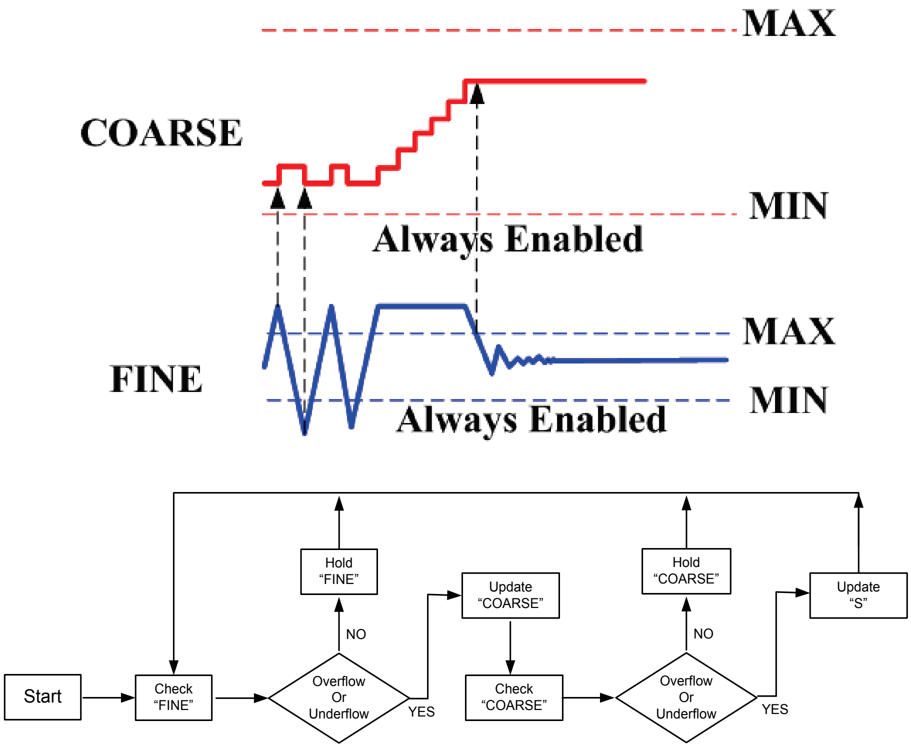


Fig. 3.12. The proposed bottom-up DCO control algorithm.

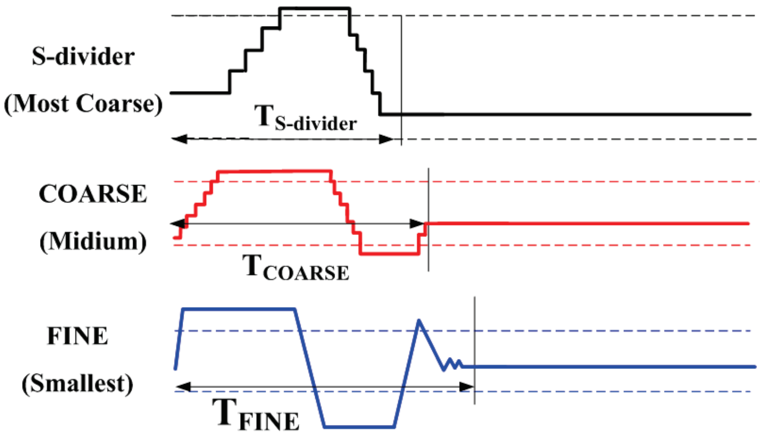


Fig. 3.13. 3-Steps bottom-up controlled DCO operation

The updating speed of the *coarse* code and S value should be controlled carefully since if the updating speed of the *coarse* code is too fast, there will be a large jump in the DCO frequency making the fast-loop unstable. To slow down the updating speed, the *coarse* control code is down-sampled by the divided clock of the reference (F_{IN}) as shown in Fig. 3.6. In similar manner, the updating speed of S is also controlled by down-sampling the S value by the divided *HSYNC* (Fig. 3.6). The range of the optimum dividing ratios (K and X shown in Fig. 3.6) are obtained from Verilog simulation, which are 4~16 for the coarse code and 2~8 for the S-value.

The upper and lower limits of the *fine* code are adaptively controlled to obtain a frequency overlap margin between the control hierarchies. When the loop is not settled, the lower/upper limits of the *fine* control move onto the near center. A lower limit is set to 25% point of the full *fine* control range while the upper limit is set to 75% point. Finally, the *coarse* code is adjusted until the *fine* code settles between 25% and 75% of the full *fine* code tuning range. After the TDC mode enters the fine mode, the min/max thresholds are extended back to 0% ~100%. The *fine* code finally settles down to have a 25% margin from the top/bottom limits. Unlike the *fine* code, the min/max limits of the *coarse* control are simply determined by the word length of the coarse control. If the word length is n-bit wide, then the minimum is 0 and the maximum is 2^n-1. However, the min/max limits of S are not fixed but dependent on the operating condition.

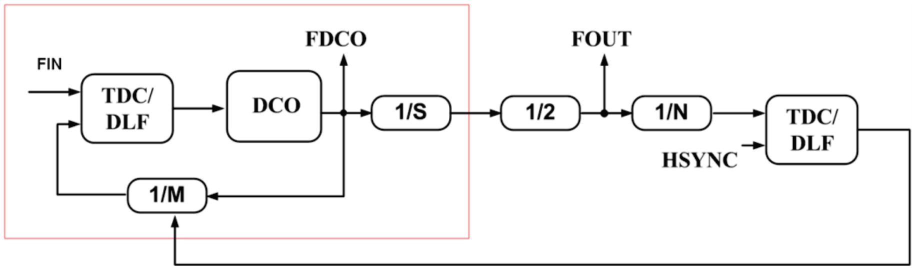


Fig. 3.14. Simplified block diagram of the proposed dual-loop PCG.

Fig. 3.14 illustrates that $\mathrm{F_{DCO}}$, $\mathrm{F_{IN}}$, $\mathrm{F_{OUT}}$ and HSYNC satisfy (3.3) and (3.4).

$$\mathrm{F_{DCO}} = \mathrm{F_{IN}} \times \mathrm{M} = \mathrm{HSYNC} \times \mathrm{N} \times \mathrm{S} \times 2 \tag{3.3}$$

$$\mathrm{F_{OUT}} = \mathrm{HSYNC} \times \mathrm{N} = \frac{\mathrm{F_{IN}} \times \mathrm{M}}{\mathrm{S} \times 2} \tag{3.4}$$

S can be derived from (3.4) as

$$\mathrm{S} = \frac{\mathrm{F_{IN}} \times \mathrm{M}}{\mathrm{HSYNC} \times \mathrm{N} \times 2} = \frac{\mathrm{M}}{\mathrm{N}} \times \frac{\mathrm{F_{IN}}}{\mathrm{HSYNC} \times 2} \tag{3.5}$$

The *S* value can be rewritten as the following.

$$S = \frac{F_{DCO}}{F_{IN}} \times \frac{F_{IN}}{HSYNC} \times \frac{1}{N \times 2} \tag{3.6}$$

The *HSYNC* and *N* are the fixed constants defined by the VESA standard [57]. F_{IN} is also a constant value defined by the external clock source. The frequency ratios like $\frac{F_{DCO}}{F_{IN}}$ and $\frac{F_{IN}}{HSYNC}$ can be measured by using a simple counter circuit shown in Fig. 3.15. The 1/2 divider in the front are added to reduce the operating speed of the counter.

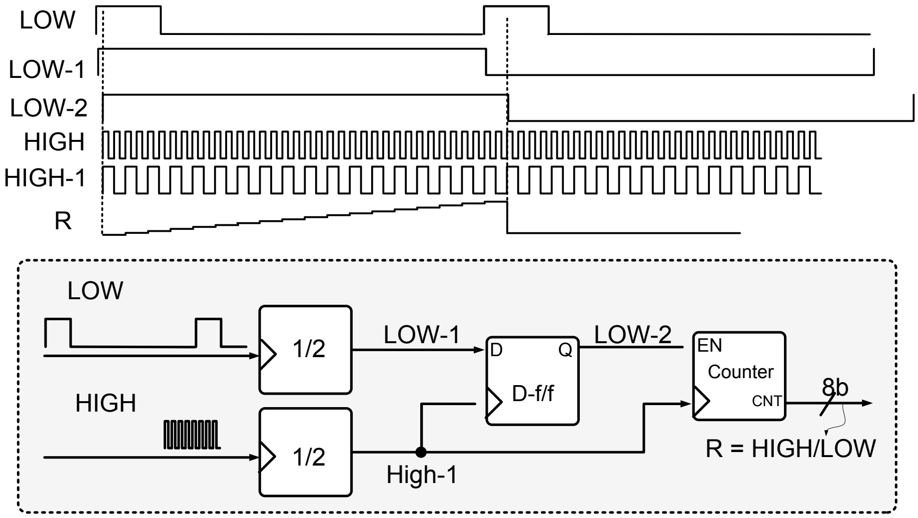


Fig. 3.15. Frequency ratio measure block.

From (3.6), the S value has only one unknown parameter, which is the DCO frequency (F_{DCO}), while others remain constant for a given video standard. Although we cannot find the exact value of the $F_{DCO,}$ the min/max values can be obtained by setting the DCO into min/max conditions. The min/max values of S can be calculated by inserting the min/max values of the F_{DCO} into (3.6), and is expressed as (3.7) and (3.8)

$$\mathrm{S_{MAX}} = \frac{\mathrm{MAX}\{F_{DCO}\}}{F_{IN}} \times \frac{F_{IN}}{HSYNC} \times \frac{1}{N \times 2} \tag{3.7}$$

$$\mathrm{S_{MIN}} = \frac{\mathrm{MIN}\{F_{DCO}\}}{F_{IN}} \times \frac{F_{IN}}{HSYNC} \times \frac{1}{N \times 2} \tag{3.8}$$

The advantage of knowing the min/max values of S is that useless iterations can be skipped while searching for the S value. If the final S value is between S_{MIN} and S_{MAX}, the DCO frequency is always within the available region to generate the target F_{OUT} satisfying (3.4).

Fig. 3.16 shows the proposed automatic divide ratio control method and the conventional approach with a fixed divide ratio. If the divide ratio is fixed, then the available frequency range is affected by PVT variations as shown in Fig. 3.16(a), whereas the proposed control algorithm provides distinct S values according to the PVT conditions. Fig. 3.16(b) shows that the S value increases with a fast process corner and decreases with a slow process corner. The available frequency range is extended using the proposed automatic S value control algorithm.

The complete block diagram of the S value calculation unit is depicted in Fig. 3.17. In order to save the power and area, the time division multiplexing technique is utilized. The ratio comparator block inputs are sequentially selected in different time slots (#1~#3) to calculate three frequency ratios (R). In the last two time slots (#4, #5), the S_{MIN} and S_{MAX} are sequentially calculated by choosing the right inputs among the stored “R” values. The calculated S values are also sequentially stored in the registers, so the hardware size is reduced by 60% using this technique that has only one multiplier, one divider, and one frequency counter.

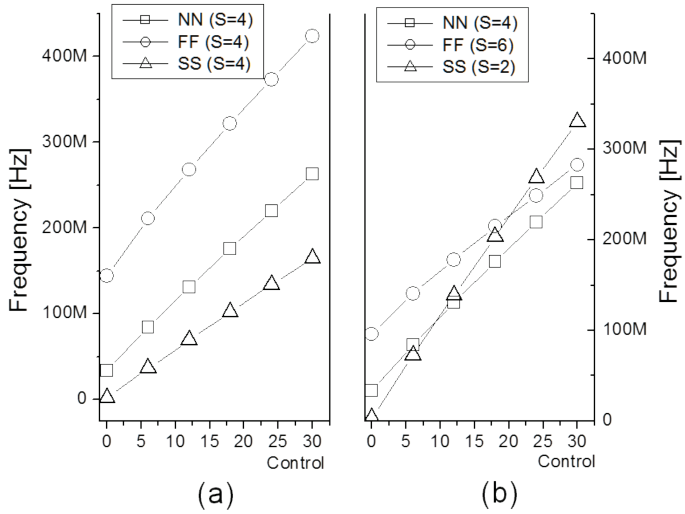


Fig. 3.16. The effectiveness of an automatic S value control. (a) The conventional fixed dividing method is severely affected by PVT variations. (b) The proposed S value control algorithm compensates PVT variations.

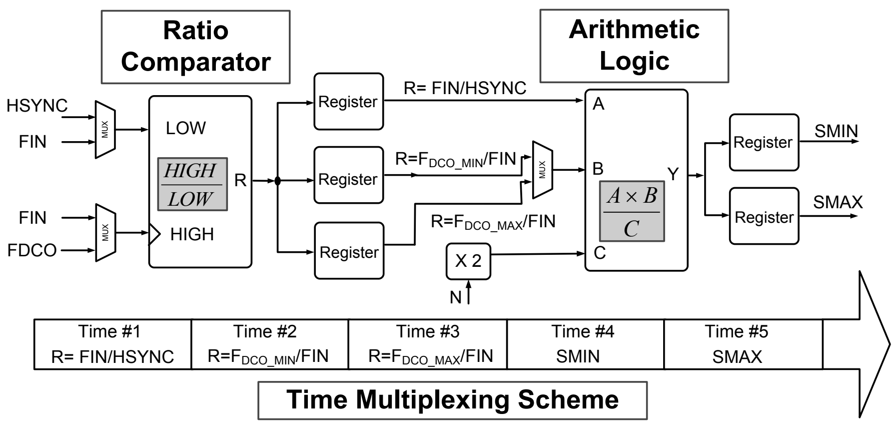


Fig. 3.17. S-range calculation block. Time multiplexing scheme is used to share the commonly used blocks.

Fig. 3.18. illustrates the operation of the s-value control block. During the initial stage of the PLL start-up, the frequency ratios R1, R2, and R3 are calculated in time division manner. After that, the S_{MAX} and S_{MIN} are

calculated. S_{MID} is the average value of the S_{MAX} and S_{MIN}. When the MIN/MID/MAX of the s-value are found, a flag signal is goes to high to show the measurement step is completed. Now, PLL starts its closed loop operation and the s-value iterates between MIN/MAX ranges and finally goes to an optimum value.

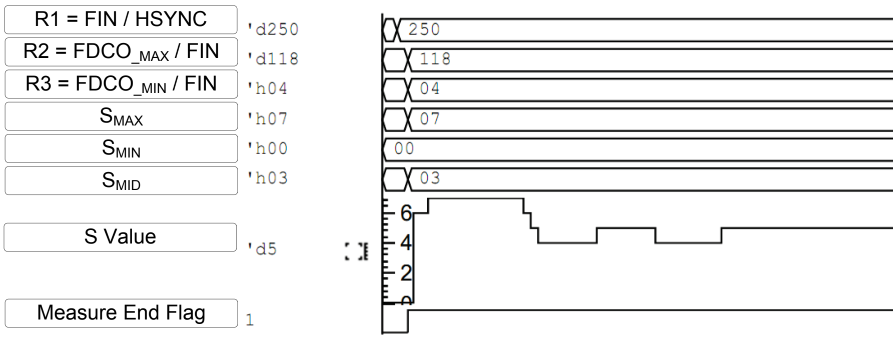


Fig. 3.18. s-value control block simulation.

3.2.4 **Digital Loop Filter**

The fast-loop operates in fraction-N mode to obtain a good frequency resolution as an equivalent DCO block. Thus, the delta-sigma modulation induced noise can be problem. DSM noise can be filtered out by decreasing the loop bandwidth, but a narrow loop BW degrades the DCO noise contribution. In this work, we have adopted a higher order loop filter as shown in Fig. 3.19. It has the cascaded 3-stage IIR filter and the conventional Proportional and Integral filter. The transfer functions of the filter are derived in the previous section 2.2.2 (page 29). The operating frequency of IIR filter is higher than the P/I filter to reduce the loop latency and to set a higher cut-off frequency of the IIR filter. The loop transfer function is determined by the product of all transfer functions of IIR and P/I filter as the following. F_{IIR} and λ are the operating frequency and the filter coefficient of the IIR filter. The α and β are proportional and integral coefficient of the P/I filter. The gain control block adjust the loop bandwidth of the P/I filter to speed up the locking process. The loop bandwidth is initially set as high value and settle down to an optimum value as the PLL approach the locked state.

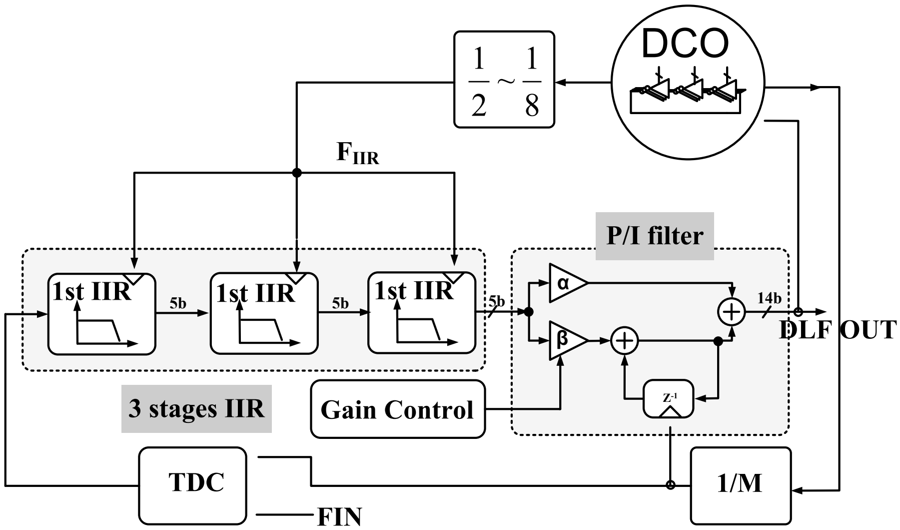


Fig. 3.19. The digital loop filter block (Gray colored part) of the fast-loop PLL.

$$Z_2(s) = \left\{ \frac{1 + \frac{s}{F_{IIR}}}{1 + \frac{s}{\lambda \cdot F_{IIR}}} \right\}^3 \cdot \left\{ \left(\alpha + \frac{\beta}{2} \right) + \frac{F_{IN} \cdot \beta}{s} \right\} \tag{3.9}$$

Unlike the fast-loop PLL, the slow-loop adopts only the P/I filter because it operates in an Integer-N PLL mode.

3.3 S-domain Noise Model

Fig. 3.20 shows the linear model of the proposed dual-loop ADPLL. The internal blocks such as the TDC and the DCO are described in the s-domain as reported in [52, 66]. The equations for the quantization noise sources are derived in the section 2. The phase noise of the input clock and the DCO are obtained from the measurements and simulation results.

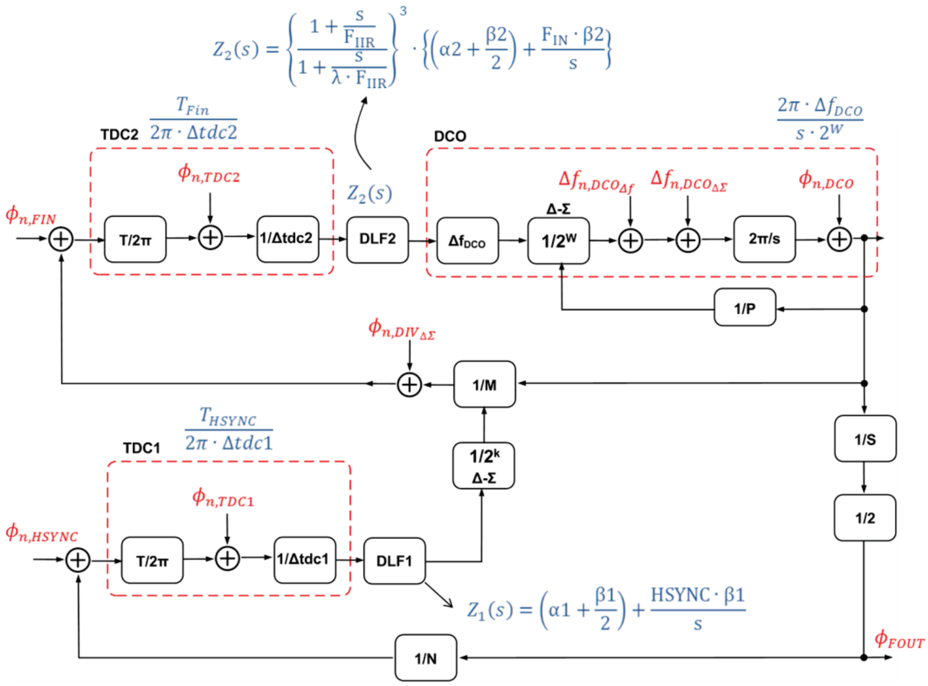


Fig. 3.20. s-domain Noise model of the dual loop ADPLL

In order to simplify the analysis, the fast-loop is independently analyzed while assuming that the divider value M is almost constant. This assumption is reasonable, because the loop bandwidth of the fast-loop is much larger than that of the slow loop and the loop filter output of the slow loop is almost constant after the entire loop settles down. In fact, the *M value* is very slowly modulated and the jitter with frequency less than the loop bandwidth of the slow loop is filtered out. The open loop gain of the fast-loop is expressed by (3.10), and the fast-loop output phase noise S_{fast} is determined by (3.12). The individual noise sources are transferred to the output by the transfer function as shown in the Table. 3.5. $Z_2(s)$ is the transfer function of the digital loop filter, which is composed of 3-rd order IIR filter and proportional/Integral filter as shown in Fig. 3.19. F_{IIR} is a sampling clock frequency and λ is a filter coefficient.

$$H_f(f) = \frac{T_{Fin}}{2\pi \cdot \Delta tdc2} \cdot Z_2(s) \cdot \frac{2\pi \cdot \Delta f_{DCO}}{s \cdot 2^W} \cdot \frac{1}{M} \tag{3.10}$$

$$Z_2(s) = \left\{ \frac{1 + \frac{s}{F_{IIR}}}{1 + \frac{s}{\lambda \cdot F_{IIR}}} \right\}^3 \cdot \left\{ \left(\alpha 2 + \frac{\beta 2}{2} \right) + \frac{F_{IN} \cdot \beta 2}{s} \right\} \tag{3.11}$$

$$S_{fast} = S_{out,FIN} + S_{out,TDC2} + S_{out,DIV} + S_{out,DCO_{\Delta\Sigma}} + S_{out,DCO_{\Delta f}} + S_{out,DCO} \tag{3.12}$$

In the aspect of the whole loop, the fast-loop functions as a DCO block with the control input *M* coming from the slow-loop and quantization noise term due to the limited frequency resolution. The fast-loop has the frequency relation determined by (3.13), where k is the word length of the DSM used for dithering the 1/M divider of the fast loop.

Thus, the fast-loop can be considered as a DCO block having the frequency resolution of $\frac{F_{IN}}{2^k}$. To find the quantization noise due to a limited DCO frequency resolution, (2.37) is modified as (3.14)

Table. 3.5. Output noise components of the fast-loop

Noise Source	Symbol	Contribution to PLL output
Input clock	$S_{out,FIN}$	$(\phi_{n,FIN}) \cdot \left\lvert M \cdot \frac{H_f}{1+H_f} \right\rvert^2$
TDC resolution	$S_{out,TDC2}$	$\frac{(\Delta tdc)^2}{12 \cdot f_R} \cdot \left(\frac{2\pi}{T_R}\right)^2 \cdot \left\lvert M \cdot \frac{H_f}{1+H_f} \right\rvert^2$
Divider DSM noise	$S_{out,DIV}$	$\left\{ \frac{(2\pi)^2}{12 \cdot f_R} \cdot \left(2 \cdot \sin\left(\frac{\pi f}{f_R}\right)\right)^{2(m-1)} \right\} \cdot \left\lvert M \cdot \frac{H_f}{1+H_f} \right\rvert^2$
DCO dithering	$S_{out,DCO_{\Delta\Sigma}}$	$\frac{1}{12} \cdot \left(\frac{\Delta f_{DCO}}{f}\right)^2 \cdot \frac{1}{f_{\Delta\Sigma}} \cdot \left\{2 \cdot \sin\left(\frac{\pi f}{f_{\Delta\Sigma}}\right)\right\}^{2n} \cdot \left\lvert \frac{1}{1+H_f} \right\rvert^2$
DCO resolution	$S_{out,DCO_{\Delta f}}$	$\frac{1}{12} \cdot \left(\frac{\Delta f_{DCO}}{f \cdot 2^w}\right)^2 \cdot \frac{1}{f_{\Delta\Sigma}} \cdot \left\{\text{sinc}\left(\frac{f}{f_{\Delta\Sigma}}\right)\right\}^2 \cdot \left\lvert \frac{1}{1+H_f} \right\rvert^2$
DCO random noise	$S_{out,DCO}$	$(\phi_{n,DCO}) \cdot \left\lvert \frac{1}{1+H_f} \right\rvert^2$

$$F_{fast} = F_{IN} \times (M_{integer} + \frac{M_{frac}}{2^k}) \tag{3.13}$$

$$\Delta f_{n,fast_{\Delta f}} = \frac{\left(\frac{F_{IN}}{2^k}\right)^2}{12 \cdot F_{IN}} \cdot \left\{\text{sinc}\left(\frac{f}{F_{IN}}\right)\right\}^2 \ \left[{Hz^2}/{Hz}\right] \tag{3.14}$$

The output phase PSD is expressed as (3.15)

$$S_{out,fast_{\Delta f}} = \left(\Delta f_{n,fast_{\Delta f}}\right) \cdot \left|\frac{1}{jf} \cdot \frac{H_{open}}{1 + H_{open}}\right|^2$$

$$= \left(\frac{\left(\frac{F_{IN}}{2^k}\right)^2}{12 \cdot F_{IN}} \cdot \left\{\text{sinc}\left(\frac{f}{F_{IN}}\right)\right\}^2\right) \cdot \left|\frac{1}{jf} \cdot \frac{H_s}{1 + H_s}\right|^2 \qquad \left[\frac{rad^2}{Hz}\right]$$

(3.15)

The open loop gain of the slow loop is expressed by (3.16) with loop filter transfer function $Z_1(s)$, and the final noise spectral density of the dual loop is written by (3.18). The individual noise components are summarized in Table. 3.6. The divider DSM noise is not included because the slow-loop is integer-N PLL. And the DSM dithering noise of DCO is already considered as the divider DSM noise $S_{out,DIV}$ during the fast-loop analysis.

$$H_s(f) = \frac{T_{Fin}}{2\pi \cdot \Delta tdc1} \cdot Z_1(s) \cdot \left(\frac{\frac{F_{IN}}{2^k} \cdot \frac{1}{2 \cdot S}}{jf}\right) \cdot \frac{1}{M}$$

(3.16)

$$Z_1(s) = \left(\alpha 1 + \frac{\beta 1}{2}\right) + \frac{HSYNC \cdot \beta 1}{s}$$

(3.17)

$$S_{out\_total} = S_{out,HSYNC} + S_{out,TDC1} + S_{out,fast_{\Delta f}} + S_{out,fast}$$

(3.18)

The output phase is calculated as the following.

$$\mathcal{L}_{out\_total}(f) = 10 \cdot \log_{10}\left(S_{out\_total}\right) \qquad \left[\frac{dBc}{Hz}\right]$$

(3.19)

Table. 3.6. Output noise components of the slow-loop

Noise Source	Symbol	Contribution to PLL output
Input clock	$S_{out,FIN}$	$(\phi_{n,HSYNC}) \cdot \left\vert N \cdot \frac{H_s}{1+H_s} \right\vert^2$
TDC resolution	$S_{out,TDC2}$	$\frac{(\Delta tdc)^2}{12 \cdot HSYNC} \cdot \left(\frac{2\pi}{T_{HSYNC}}\right)^2 \cdot \left\vert N \cdot \frac{H_s}{1+H_s} \right\vert^2$
Fast-loop resolution	$S_{out,fast_{\Delta f}}$	$\frac{1}{12} \cdot \left(\frac{F_{IN}}{f \cdot 2^k}\right)^2 \cdot \frac{1}{F_{IN}} \cdot \left\{ \mathrm{sinc}\left(\frac{f}{F_{IN}}\right) \right\}^2 \cdot \left\vert \frac{1}{1+H_s} \right\vert^2$
Fast-loop Noise	S_{fast}	$(S_{fast}) \cdot \left\vert \frac{1}{1+H_s} \right\vert^2 \cdot \left(\frac{1}{2 \cdot S}\right)^2$

Table. 3.7 summarizes the loop parameters in Fig. 3.21 and Fig. 3.22. Unless commented otherwise, we used the values given in Table. 3.7. Fig. 3.21 shows the linear analysis results for the fast-loop where the open-loop gain is denoted as $A_{fast}(f)$ and the closed-loop transfer function is shown as $G_{fast}(f)$ *in* Fig. 3.21(a). The non-DCO noise components are shown in Fig. 3.21(b) which follow a low-pass transfer function $G_{fast}(f)$. On the other hand, the DCO-related noise sources follow a high-pass transfer function $1\text{-}G_{fast}(f)/M$ as presented in Fig. 3.21(c). The overall phase noise of the fast-loop is calculated by summing the transferred noise components of (b) and (c), and is illustrated in Fig. 3.21(d).

Table. 3.7. Loop Parameter for s-domain analysis

TDC Resolution		Dither Freq	DSM order	
Δt_{TDC1}	Δt_{TDC2}	$f_{\Delta\Sigma}$	$n_{\Delta\Sigma\_DCO}$	$n_{\Delta\Sigma\_DIV}$
20ps	60ps	375e6	1	3
Loop Bandwidth			DSM Word length	
f_{BW_SLOW}	f_{BW_FAST}	f_{IIR_3dB}	w	k
10kHz	1.5MHz	7MHz	6	16

Δt_{TDC1} : **TDC resolution of the slow loop**
Δt_{TDC2} : **TDC resolution of the fast loop**
$f_{\Delta\Sigma}$: **Dithering frequency of ring DCO**
$n_{\Delta\Sigma\_DCO}$: **The order of DCO dithering DSM**
$n_{\Delta\Sigma_{DIV}}$: **The order of the DSM divider in the fast loop**
f_{BW_SLOW}: **Loop bandwidth of slow loop**
f_{BW_FAST}: **Loop bandwidth of fast loop**
f_{IIR_3dB}: **3dB frequency of IIR filter** of the fast loop
w: Word length of DCO dithering DSM
k: Word length of main divider (M) DSM

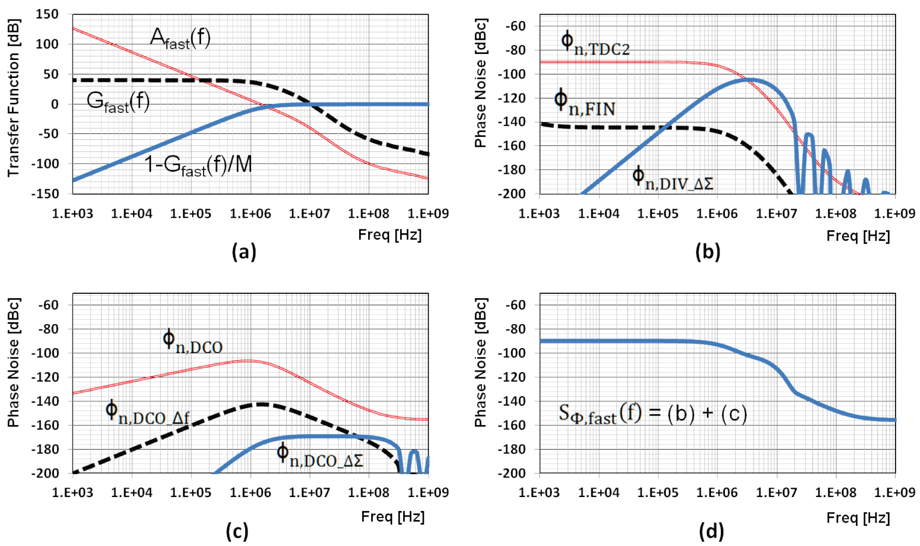


Fig. 3.21. s-domain noise analysis for the fast loop. (a) Loop transfer functions, (b) the output noise due to various noise components, (c) the output noise due to DCO-related components, and (d) the total phase noise of the fast loop. The inserted table shows the key loop parameter.

The analysis of the slow loop is presented in a similar manner in Fig. 3.22. There is no delta-sigma noise component because the slow loop is an integer-N PLL. The noise sources of Fig. 3.20 are multiplied by the loop transfer functions of Fig. 3.22(a), and the noise components transferred to the output are plotted in Fig. 3.22(b) and (c). The total phase noise is obtained by combining all of the noise outputs of Fig. 3.22(b) and (c), which is shown in Fig. 3.22(d).

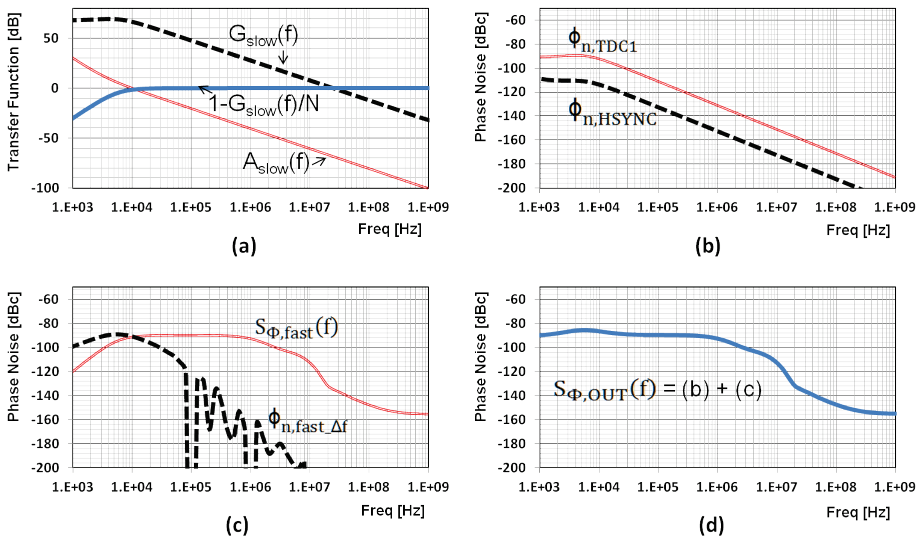


Fig. 3.22. s-domain noise analysis for the slow loop. (a) Loop transfer functions, (b) the output noise due to input various noise components, (c) the output noise due to the fast-loop PLL related components, and (d) the total phase noise of the slow-loop. The phase noise is dominated by the fast-loop PLL related components (c) since the loop BW is extremely low.

3.4 Loop Parameter Optimization Based on the s-domain Model

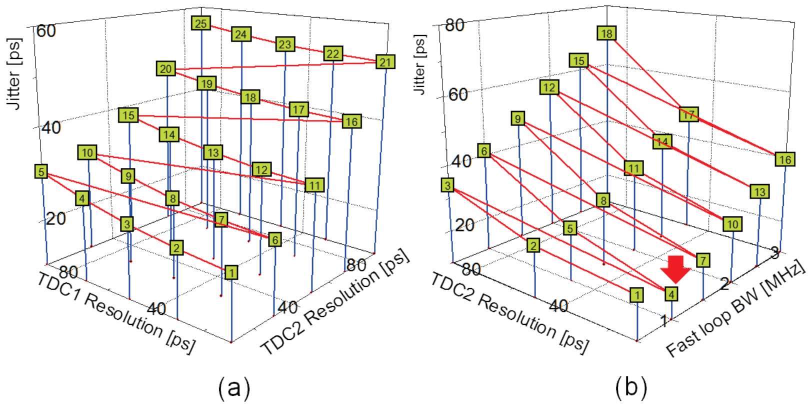


Fig. 3.23. The effect of TDC quantization noise. (a) While both TDC1 and TDC2 affect the jitter of the final output of the dual-loop PLL, the effect of TDC2 is more dominant. (b) The jitter degradation, due to the limited TDC resolution (10ps ~ 90ps), is proportional to the loop bandwidth (0.5MHz ~ 3MHz).

Fig. 3.23(a) shows that the limited TDC resolution degrades the over-all jitter performance as derived in (2.33). The effect from TDC2 is more dominant due to its wide loop bandwidth; TDC noise follows a low pass transfer function as in Table. 3.5 and Table. 3.6. Fig. 3.23(b) proves that the quantization noise contribution is proportional to the loop bandwidth. In conclusion, if the loop bandwidth is large, then we should improve the TDC resolution to suppress the in-band noise. If the loop bandwidth is narrow then the TDC resolution requirement can be relaxed. However, point “4” of Fig. 3.23(b) shows that we need a trade-off to obtain a minimum jitter value. When the TDC resolution is small enough, the overall jitter performance is dominated by the DCO random noise and DSM quantization noise.

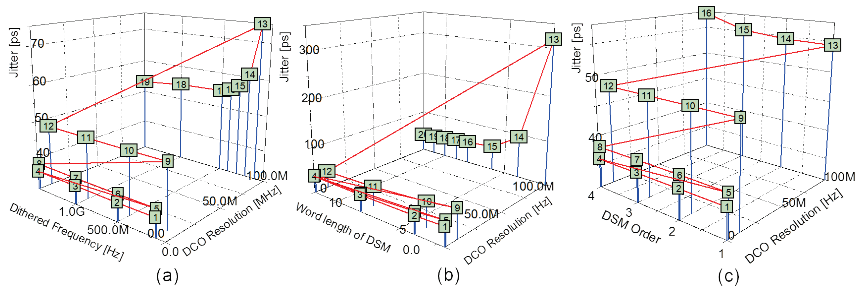


Fig. 3.24. The effect of DCO quantization noise. (a), (b) the dithering frequency (0.1GHz~1.5GHz) and the DSM word length (2~12) affect the jitter value only when the DCO resolution is coarse. (c) The DSM order (1~4) does not significantly change the jitter value.

Fig. 3.24 illustrates the effect of the DCO quantization noise. Equation (2.38) note that the DCO quantization noise can be reduced by increasing the dithering speed and the DSM word length. A higher order DSM is more helpful in randomizing the control sequence. However, Fig. 3.24(a) and (b) show that the dithering speed and word length are only meaningful with a coarse resolution. In this design, the DCO has a fairly good resolution of 1 MHz, thus the impact of the DSM performance is insignificant. Fig. 3.24(c) shows that 1^{st} order DSM is sufficient, because the phase noise of the ring DCO overwhelms the DCO quantization noise across the spectrum, as shown in Fig. 3.21(c). Based on the s-domain analysis results, a 1^{st} order DSM with 6-bit input and 375-MHz dithering frequency is adopted. This conclusion of Fig. 3.24(c) is valid only for noisy ring DCOs. The dithering speed is determined by the VCO frequency and the available dividing ratio of the post-scaler. The spur due to the 1^{st} order DSM is automatically mitigated because the loop filter outputs are actually a pseudorandom signal.

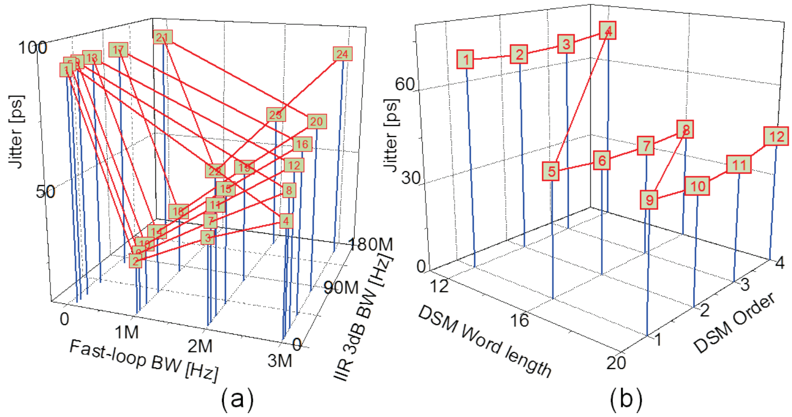


Fig. 3.25. The effect of main-divider DSM quantization noise. (a) Jitter is lowest around the 1MHz loop bandwidth. The dependency on the fast-loop BW is decreased as the IIR 3dB frequency decreases. (b) The DSM word length should be large enough to guarantee a frequency resolution of the fast-loop PLL. The DSM order has a small impact because high frequency DSM noise is suppressed by the IIR filter.

As the loop bandwidth is increased, the in-band noise coming from the DCO is reduced. However, the out-band noise due to the DSM is significantly increased, degrading the overall jitter performance. A 3-rd order IIR filter, made by three 1st order IIR filters in series, is adopted to suppress the out-band DSM noise. Fig. 3.25(a) shows the advantages of IIR filtering and as the 3dB frequency of the IIR filter is decreased, the overall jitter is significantly reduced. We can observe important trends. First, there is an optimum point with minimized jitter, which is shown in Fig. 3.25(a) around 1MHz. Second, the IIR filter mitigates the loop-bandwidth dependency by suppressing the out-band DSM noise. And third, the noise performance of the dual loop is highly affected by the fast-loop PLL because the output of the fast-loop PLL is almost completely transferred to the output of the slow loop due to its extremely low cut-off frequency (1 kHz-10 kHz). Now we should determine the word length and the order of the DSM. Fig. 3.25(b) shows that the order does not matter. This is because the DSM noise can be fully suppressed by IIR filters. The word length should be large enough for a good frequency resolution as indicated by (3.15). However, the s-domain analysis shows that word length larger than 16 bits is excessive. In this work, we utilize the 3rd order DSM to prevent a deterministic spur by randomization.

3.5 RTL and Gate Level Circuit Design

3.5.1 **Overview of the design flow**

The first step is to define the design specification of the sub-blocks like TDC, DCO, and DLF, ect. Based on the defined design target, the functional blocks except TDC and DCO are described in RTL. The TDC and DCO are written in ideal behavioral level to execute verilog simulation. And simultaneously, the unit cells of the TDC and DCO are designed manually. The unit-cells are prepared in a standard design kit format to be compatible with a digital design flow. If the behavioral simulation result is acceptable then a gate level synthesis is preceded. In this step, the behavioral code for the TDC and DCO are replaced with the gate level netlists written by a designer. After the whole gate netlist is generated, the auto P&R is executed. The design kits of the TDC and DCO are utilized during auto P&R. The over-all design flow is illustrated in the Fig. 3.26.

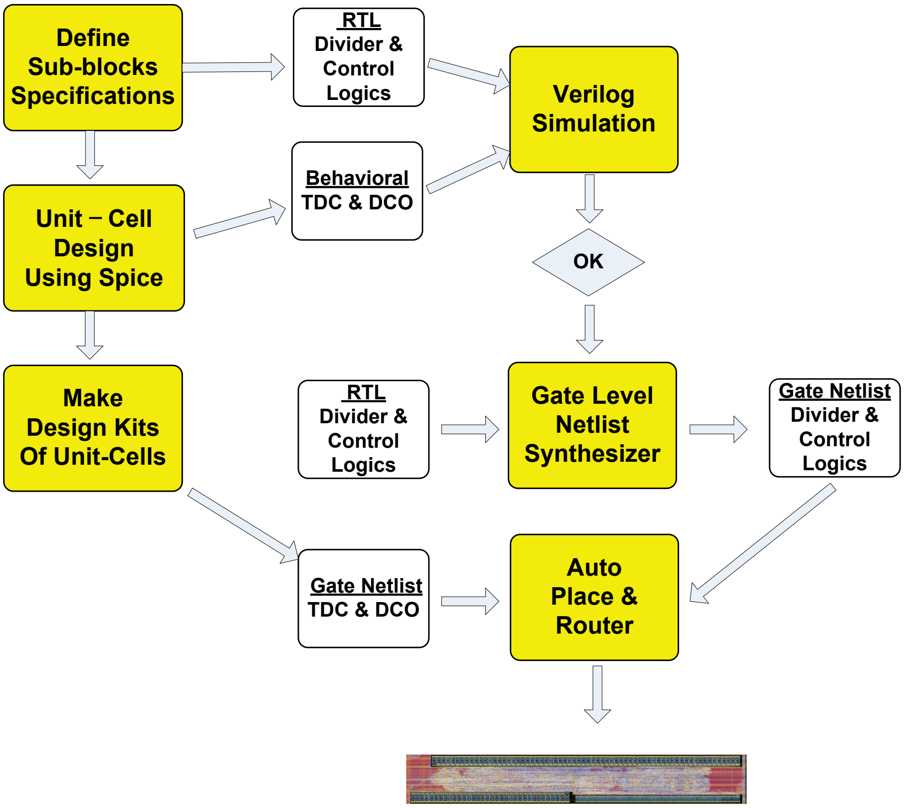


Fig. 3.26. Design flow

3.5.2 Behavioral Simulation and Gate level synthesis

The functional blocks should describe in RTL (Register transfer level) or Gate level to complete the layout using auto P&R. In this work, we described whole circuit in RTL except the TDC and DCO. The TDC and DCO cannot be synthesized from a RTL code because verilog does not support the architecture of the ring oscillator feedback and parallel connected programmable delay stages of TDC. Thus, TDC and DCO are more easily written in gate level rather than the RTL.

Even though the meta-stability has only a negligible effect in physical world, it makes a big problem during the simulation because CDC issue generates an unknown state and stops the simulation. To prevent a simulation from stopping unexpectedly, we adopt an ideal TDC and resampler during behavioral simulation. Afterward, the ideal behavioral modeling blocks are replaced with real gate level model to synthesize a layout.

Another simulation failure comes from the tri-state inverter of the TDC and DCO. The verilog cannot model the phenomenon that the driving strength of the hard wired tri-state inverter changes according the enabled tri-state inverter cells. To solve this problem, the delay stage of the TDC is described in ideal behavioral model and the whole DCO ring oscillator core is also described in behavioral level during the simulation. The ideal models are replaced with the gate level netlist during Auto P&R.

3.5.1 **Preventing a meta-stability**

Fig. 3.27 depict the frequency domains of the proposed dual-loop ADPLL. There are three clock domains, and a meta-stability problem exists at the clock domain crossing interface [20]. The meta-stability due to CDC (Clock Domain Crossing) can be solved by resampling the incoming signal by the receiving clock domain. Fig. 3.28 show how the clock domain changes after inserting resamplers at the CDC interfaces. The clock domains are changed to the F_{DCO} domain except the TDCs. Although the TDC still has a meta-stability problem, the effect is negligible because the output of the TDC is a thermometer code. The thermometer code has inherently only 1 bit error, thus it does not affect significantly for the final binary code TDC output.

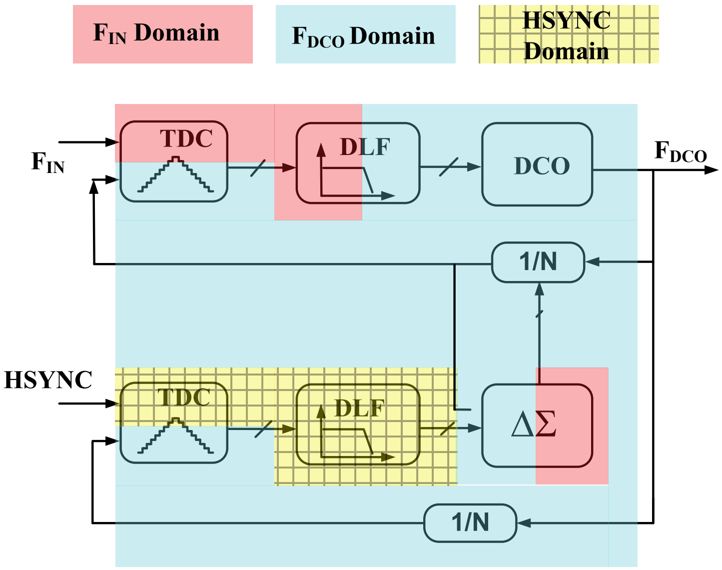


Fig. 3.27. Frequency domains of the dual-loop ADPLL

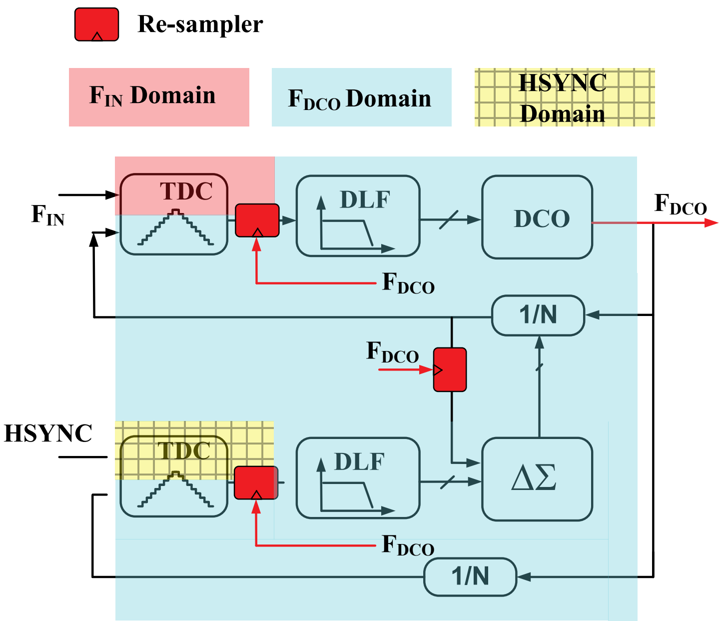


Fig. 3.28. Re-sampler insertion to prevent CDC problem.

3.5.1 **Reusable Coding Style**

The functional blocks of the ADPLL are described in HDL. To improve the reusability, we write the HDL code in a programmable format. Fig. 3.29 presents an example coding style for a TDC of which input/output bit-width is programmable. When the top module "ws_tdc" instantiates the sub-module "ws_tdc_2stg" , the bit-width is defined as the parameter "wf", "wc", and "wout". In a similar manner, the other sub modules also defined when they are instantiated by the higher hierarchy module. In addition to the bit-width, the number of recalled instances can be programmable as shown in the statement for instantiating the "ws_tdc_unit". The number of unit delay cells is determined by the parameter "wth" , which is calculated from the statement of "parameter wth=2**(wf-1)-1". Using this technique, we can easily reuse a unit functional blocks.

```
 module ws_tdc (hsyn, fdiv, fvco, rstb,tdcout)
         parameter wf=7;
         parameter wc=11;
         parameter wout=21;
         input hsyn,fdiv,fvco,rstb;
         output signed [wout-1:0] tdcout;
         ws_tdc_2stg #(wf,wc,wout) xtdc (hsyn,fdiv,fvco,rstb,tdcout);
endmodule

module ws_tdc_2stg (hsyn,fdiv,fvco,rstb,tdcout);
         parameter wf=7; // fine TDC bitwidth -31 ~ +31
         parameter wc=11; // coarse TDC bitwidth -2047 ~ 2047
         parameter wout=21; // TDC output bitwidth
         input hsyn,fdiv,fvco,rstb;
         ws_tdc_fine #(wf) xtdc (hsyn,fdiv,rstb,tdcout_fine);
         ws_tdc_coar #(wc) xtdc (en,hsyn,fdiv,fvco,rstb,tdcout_coar);
         wire signed [wout-1:0] tdcout = coar_en==1)?tdc_coar:tdc_fine;
endmodule

module ws_tdc_fine (fref,fdiv,rstb,tdcout);
         parameter wf=5;
         parameter wth=2**(wf-1)-1;
         input fref,fdiv,rstb;
         output [wf-1:0] tdcout;
        ws_tdc_unit xdly_chain [wth-1:0] (.rstb(rstb), .fref(fref),
                                          .fdiv(fdiv[wth-2:0],
                                          .fdiv_dly(fdiv_dly[wth-1:1),
                                          .dout(dout_th[wth-1:0));
         ws_ther_to_bin #(wf) xthermometer (dout_th,tdcout);
endmodule

module ws_tdc_coar (en,fref,fdiv,fvco,rstb,tdcout);
         parameter wc=5;   // TDC binary code output bitwidth
         input fref,fdiv,fvco,rstb;
         output [wc-1:0] tdcout;
         // detailed code is skipped
         ....

endmodule
```

Fig. 3.29. Reusable coding style example

3.6 Layout Synthesis

3.6.1 **Auto P&R**

While the ADPLL can be described in HDL (Hardware Description Language), the layout design has relied on the conventional custom layout drawing [70, 71]. The main reason layout synthesis is not used is due to the linearity degradation from the uncertainty during automatic Placement and Routing (auto P&R). Recently, a fully synthesized PLL has been reported [72]. They described the entire circuit in HDL and completed the layout using conventional auto P&R tools. In order to mitigate the mismatch effects of auto P&R, they rearranged the control sequence for the delay cells according to the measured driving strength. While they have succeeded in obtaining a coarse/fine control under the secondary effect of random distribution, the linearity degradation due to the uncertainty in the layout was not solved. Moreover, the resolution and the tuning range are unknown until the place and routing are completed. A robust, reliable unit-cell design technique is proposed to prevent a systematic mismatch due to the uncertainty in the conventional auto P&R process [65].

Fig. 3.30 compares two approaches using the conventional primitive cell and the proposed plug-in unit cell. Fig. 3.30(a) shows the layout from a conventional approach that connects the unit cells with an auto P&R tool. The metal lines are randomly routed and the irregular routing lines make a systematic mismatch in the DCO characteristics. Unlike the conventional method, the proposed design includes the routing path within the unit cell. Furthermore, there is no external routing path because the unit cells are connected to each other by butting. This means that the unit cells are *plugged in* side by side. A simple script or GUI-based command can be used to align the unit cells in the regular form as shown in Fig. 3.30(b).

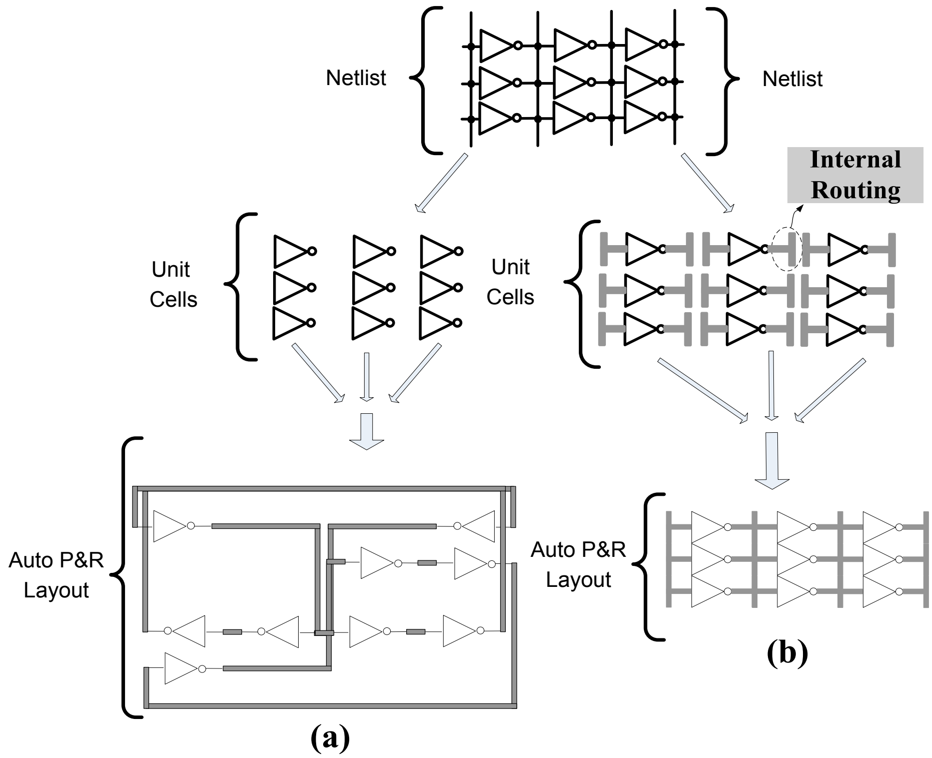


Fig. 3.30. Cell-based layout techniques. (a) The conventional auto P&R and (b) proposed plug-in cell-based technique.

Fig. 3.31 shows the layout example of the TDC and DCO which is drawn automatically by the proposed cell based technique. The plug-in unit cells are repeatedly inserted to make the whole circuit.

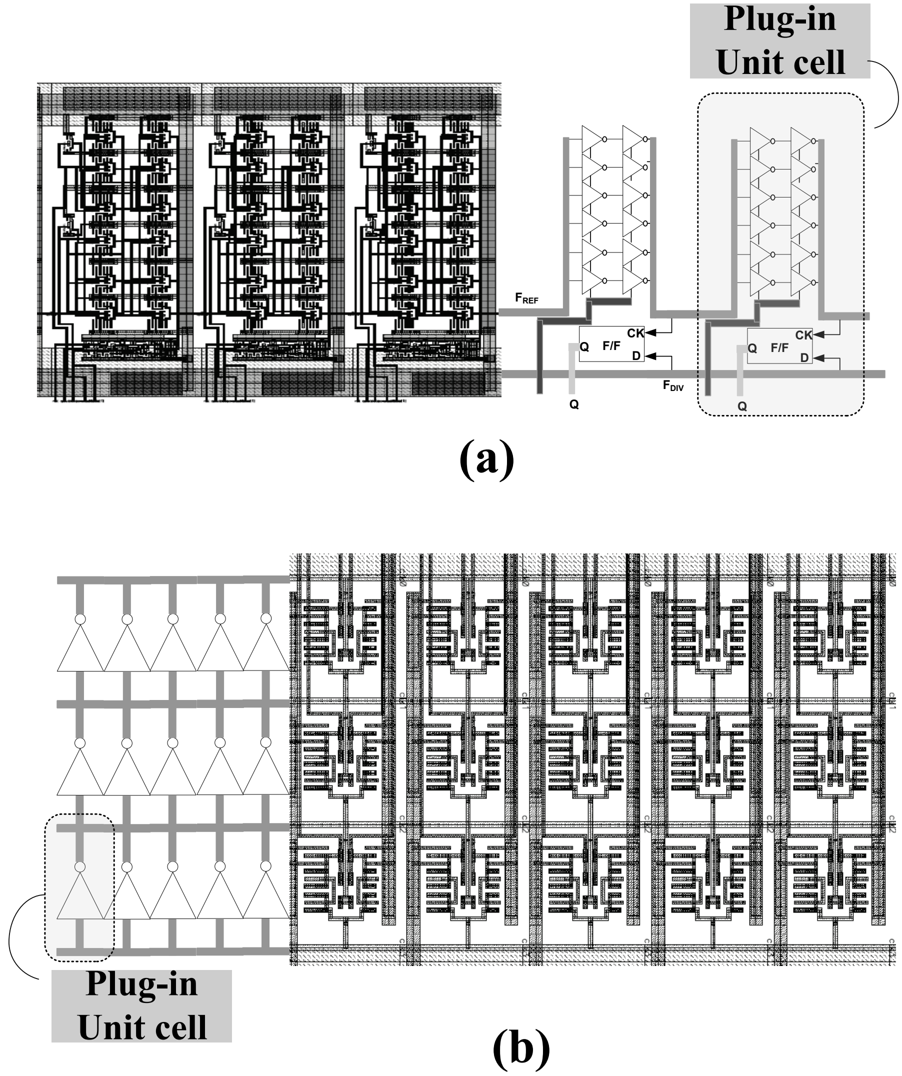


Fig. 3.31. Auto P&R using the cell based layout technique. (a) TDC, (b) DCO.

The remaining logic parts such as the divider, the loop filter and the other control logic functions are automatically placed and routed based on the given design constraints. The entire PLL layout is presented in Fig. 3.32. After the RTL code is fixed, the PLL layout is completed in less than 4 hours by utilizing the conventional auto P&R tool.

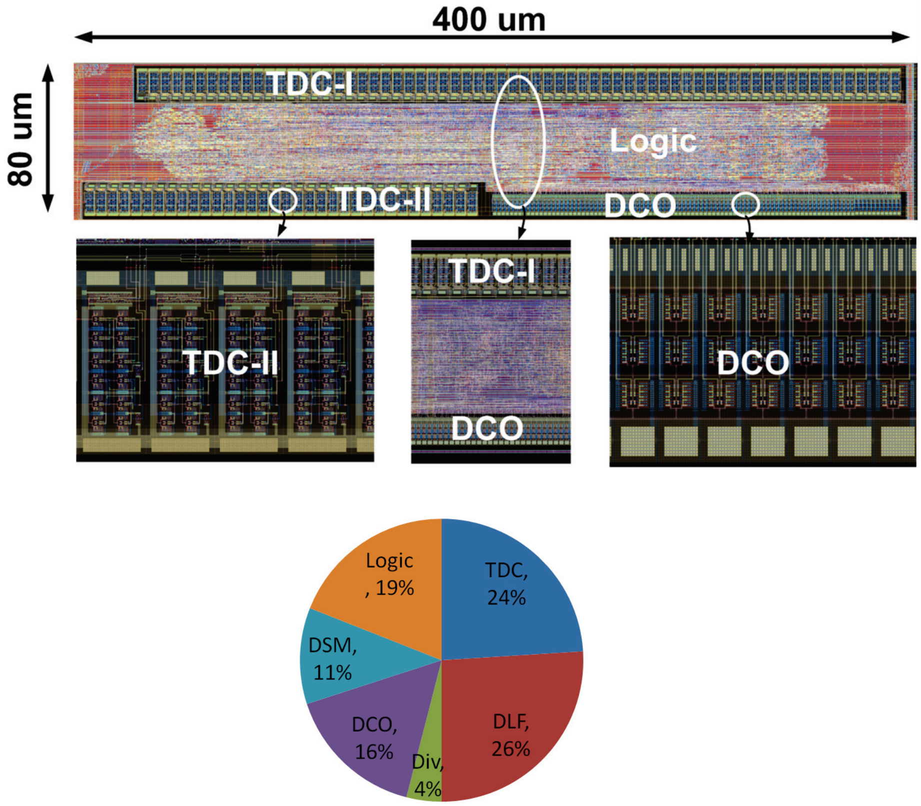


Fig. 3.32. Synthesized PLL layout. It occupies 400um x 80um.

3.6.2 **Design of Unit Cells**

We add only two tri-state inverter cells into a standard cell library. The transistor size of the unit tri-state inverter is determined from the SPICE simulations to meet the DCO resolution target and tuning range. After completing the circuit design, the layout is done manually, and the internal routings are also included so that each cell is easily connected to each other without external routings. These plug-in cells are incorporated in the standard design kit format for compatibility with the commercial auto P&R tools. The tri-state cells comprise the unit stages of the DCO and the TDC.

The unit block of ring oscillator is implemented by connecting three tri-state inverter cells in series using an auto P&R tool. There is no uncertainty in the critical signal path, because all routes are already included in the tri-state inverter cell. The unit DCO stages are connected in parallel to meet the frequency tuning range; there is no external routing.

The unit stage of the TDC is composed of the parallel connected tri-state inverters and a single D flip-flop while the resolution is determined by the number of parallel connected tri-state inverters. The SPICE simulation is used to characterize the TDC resolution and the layout is completed using auto P&R tools. The detection range is determined by the number of TDC unit stages.

3.6.3 **Linearity Degradation in Synthesized TDC**

Fig. 3.33 illustrates a unit stage of TDC and its equivalent circuit including parasitic loading components. Using Fig. 3.33, the TDC can be drawn like the Fig. 3.34. The linearity is determined by two clock of sampling F/F, therefore the TDC equivalent model can be more simplified as shown in Fig. 3.35. If the parasitic components are ignored then the TDC has completely linear characteristics. However, the rising edge of B[62:0] is not converged to a single position but distributed due to R/C delay mismatch as shown in Fig. 3.36.

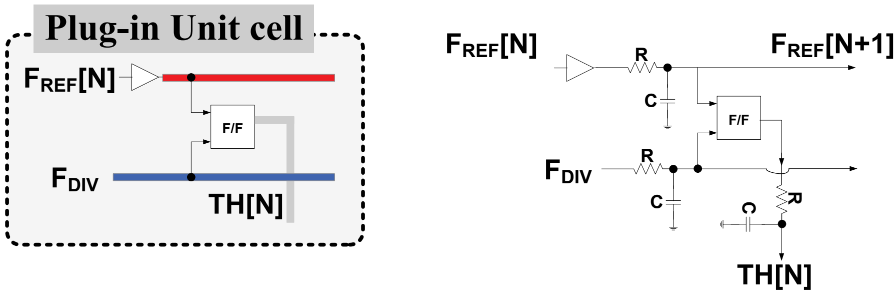


Fig. 3.33. TDC unit cell and its equivalent circuit model.

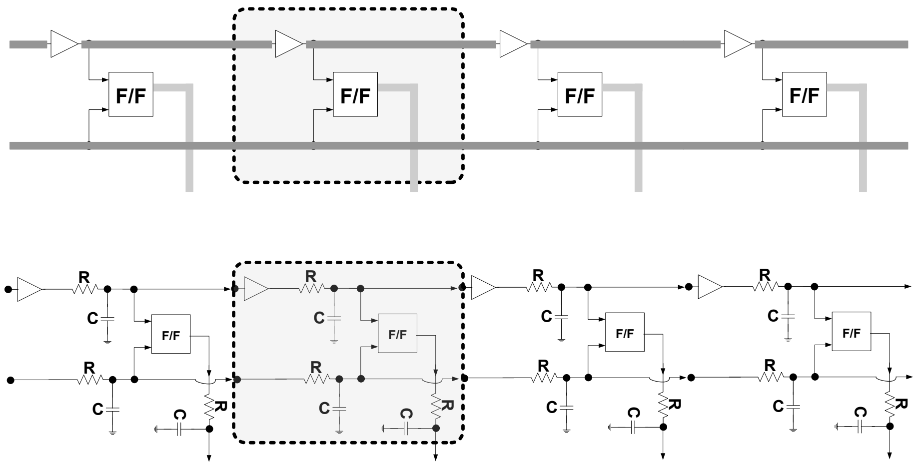


Fig. 3.34. Equivalent model of the TDC

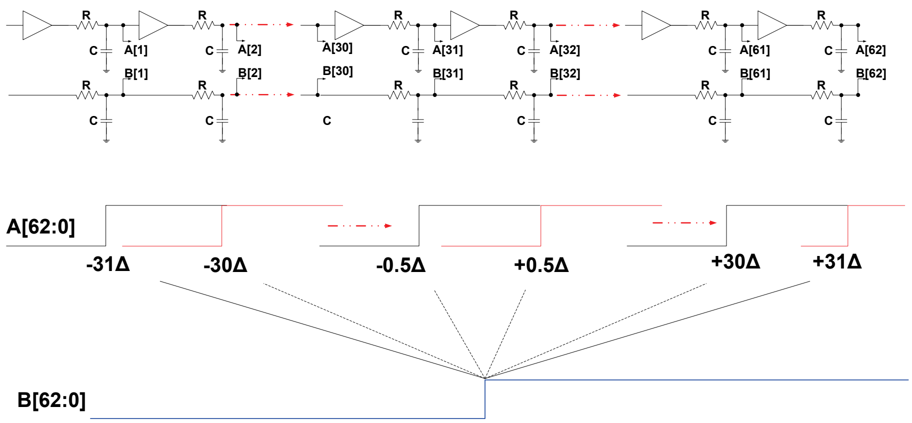


Fig. 3.35. TDC timing (When R*C=0).

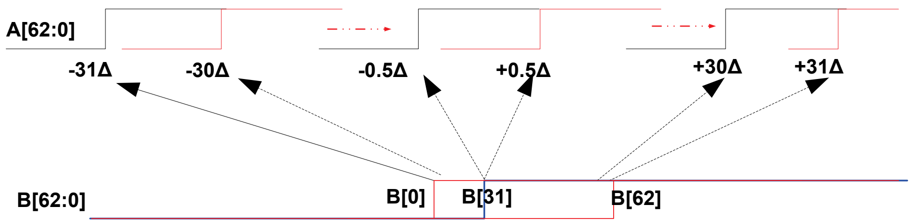


Fig. 3.36. TDC timing (When R*C≠0)

Fig. 3.37 depicts the delay distribution of B[62:0] when the peak to peak timing delay between B[0] and B[62] is 200ps. The TDC output is defined by the timing difference between A[N] and B[N], thus the resultant TDC transfer curve is degraded as shown in Fig. 3.38. The INL/DNL characteristics are plotted in Fig. 3.39. The peakings of INL/DNL near the zero timing skew come from the finite TDC resolution, thus it should be ignored.

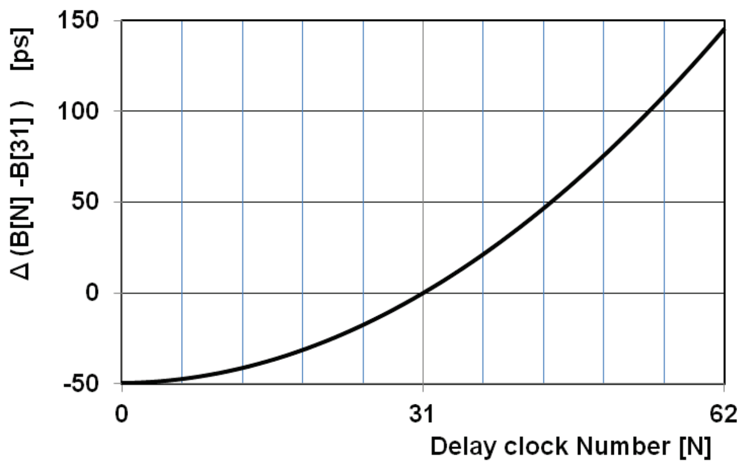


Fig. 3.37. Timing skew due to R/C delay

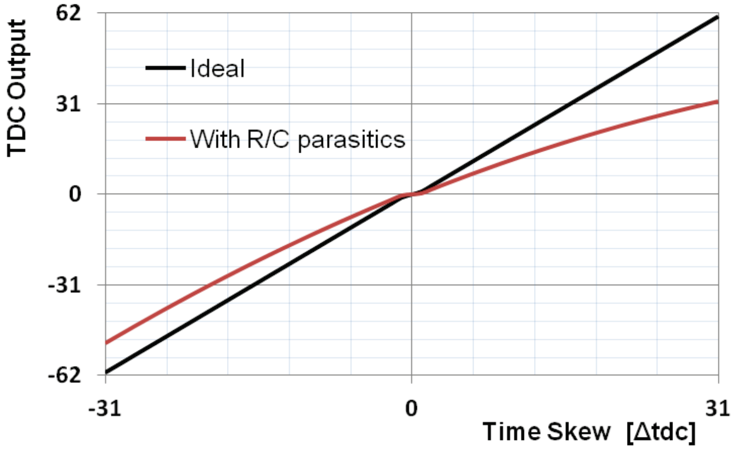


Fig. 3.38. Linearity degradation due to R/C parasitic.

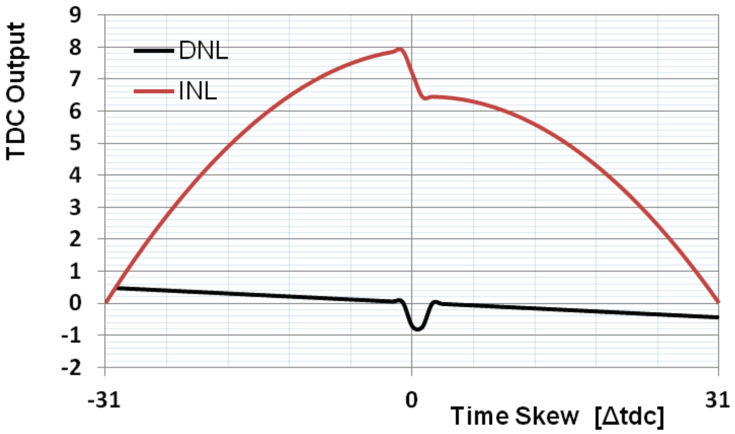


Fig. 3.39. INL/DNL degradation due to R/C delay. The peaking near the zero skew is not due to linearity degradation but due to the finite TDC resolution

Fig. 3.35 and Fig. 3.36 indicate that the linearity can be improved by reducing the delay mismatch of the B[62:0] clock path. The straight forward solution is to increase the driving buffer strength as shown in Fig. 3.40. If we reduce the whole clock path delay from 200ps to 60ps as depicted in Fig. 3.41. The linearity is enhanced from Fig. 3.38 to Fig. 3.42. Compared to the original

TDC linearity characteristics of Fig. 3.39, the large clock driver reduces the INL error by 70%, and the DNL by 50% as presented in Fig. 3.43.

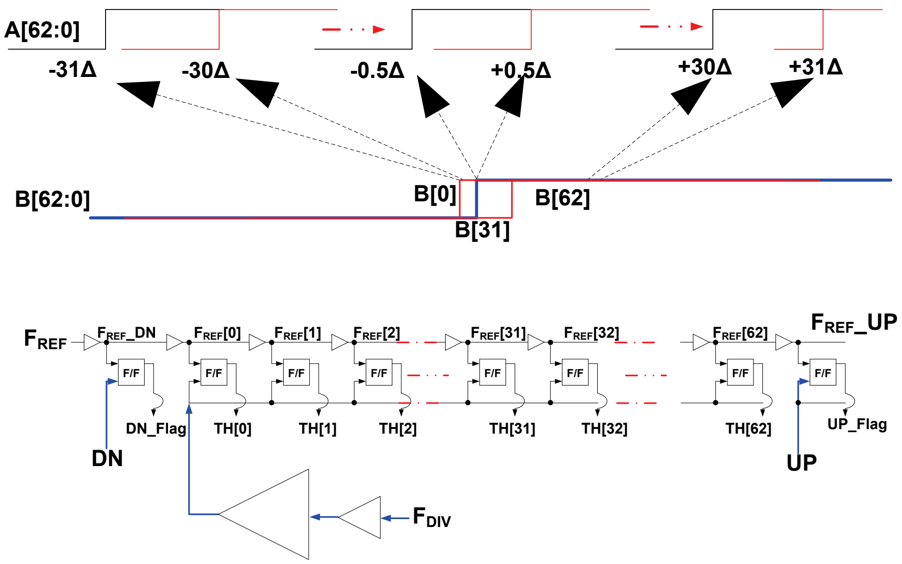


Fig. 3.40. R/C parasitic effect mitigation technique using large driving buffer. The parasitic R/C components has less effect for large driving source.

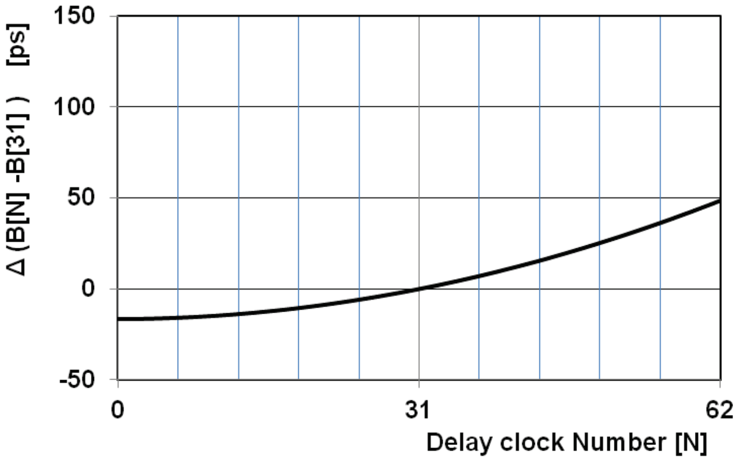


Fig. 3.41. Reduction of peak-to-peak timing skew due to large driving buffer. Assuming that the peak-to-peak timing skew between B[0] and B[62] is reduced from 200ps to 60ps

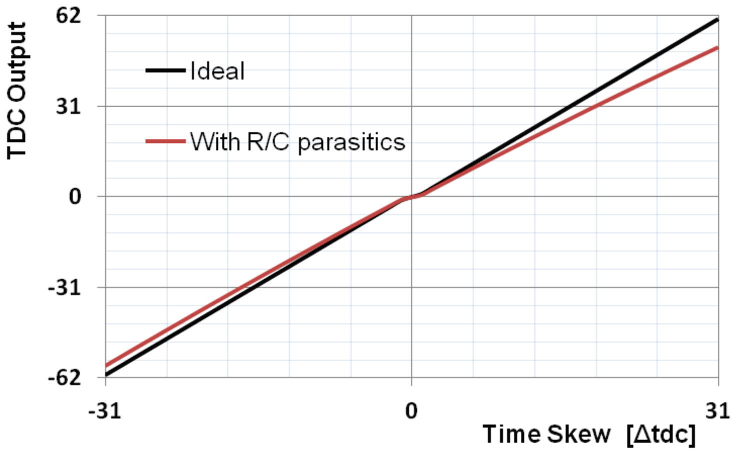


Fig. 3.42. Linearity improvement by increasing the driving strength of the buffer. Compared to the original curve (Fig. 3.38), the TDC response is more close to the ideal case.

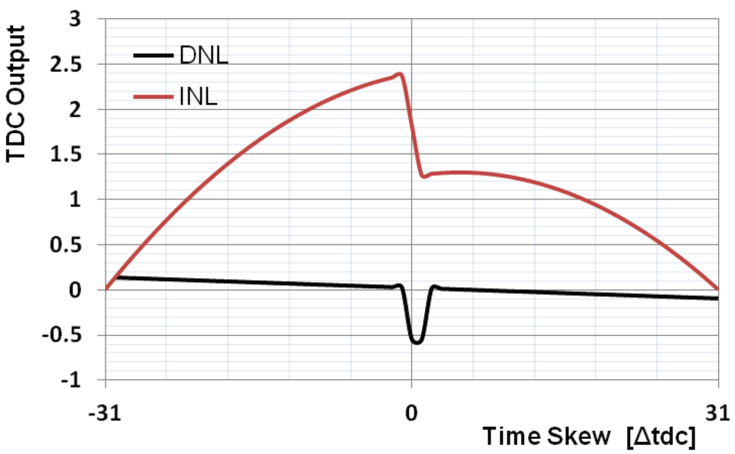


Fig. 3.43. INL/DNL improvement due to large clock buffer.

Fig. 3.44 illustrates another technique to reduce the delay mismatch, which use a symmetrical clock distribution path having a tree shape. If the clock tree is perfectly matched then there is zero timing skew between the arriving clock signals, thus the clock B[62:0] are converged to a single point. However, there is an avoidable random mismatch, and we should consider the delay variation due to random mismatch effect between clock tree paths. Fig. 3.45 presents delay variations among B[62:0] when there is +/- 0.1 LSB random mismatch. Unlike the previous single clock path schemes, the distribution of the rising edges of B[N] is random. Fig. 3.46 shows the TDC characteristic enhanced by the technique presented in Fig. 3.44. The INL is reduced by 90% compared to Fig. 3.39, and the DNL is comparable to Fig. 3.43.

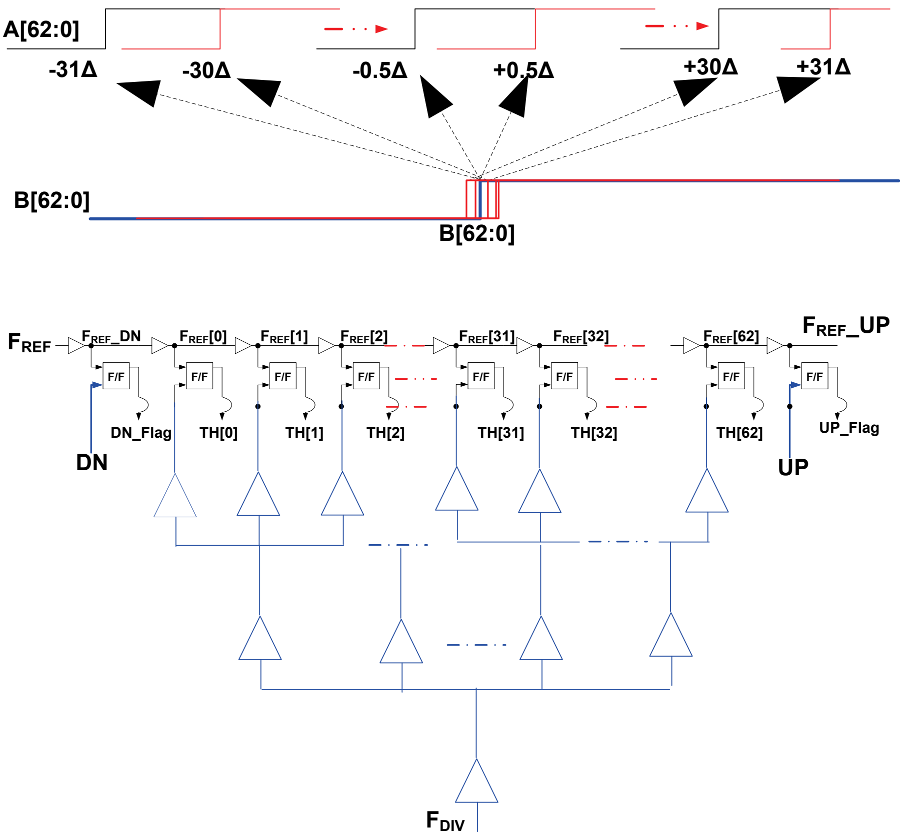


Fig. 3.44. Clock delay mismatch mitigation using a symmetrical clock distribution network.

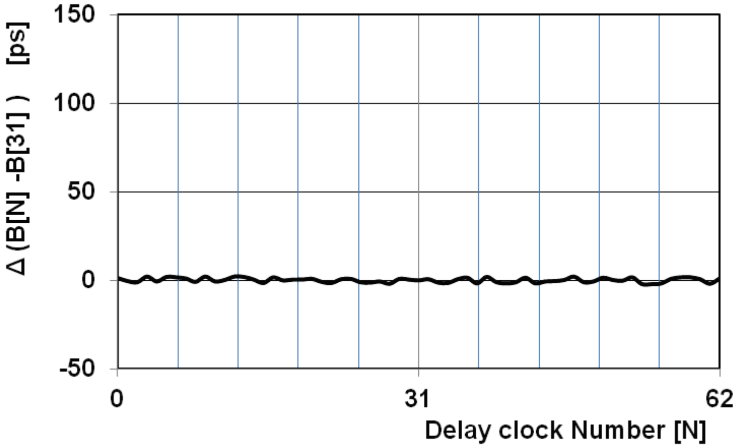


Fig. 3.45. Clock delay mismatch reduction using symmetrical clock tree.

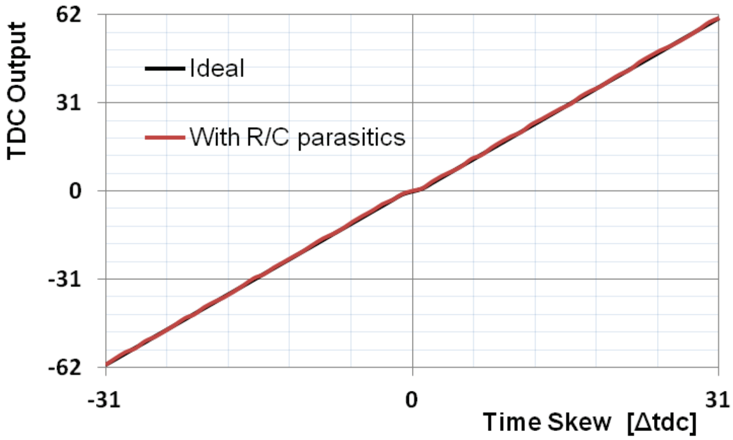


Fig. 3.46. Linearity improvement using symmetrical clock tree.

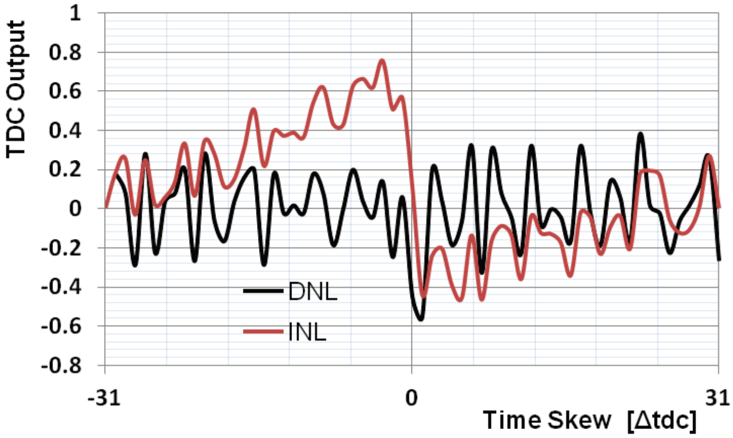


Fig. 3.47. INL/DNL improvement using symmetrical clock distribution.

3.6.4 **Linearity Degradation in Synthesized DCO**

The nonlinearity of the DCO mainly comes from the voltage dependency of the loading capacitance and the parasitic resistance along the internal routing path. In order to simplify the analysis, the parasitic resistance effect is ignored and only the capacitance nonlinearity is analyzed. The 8-bit fine tuning characteristic of a 3-stage ring oscillator is determined approximately by (3.20)

$$F_{DCO} = \left(\sum_{n=1}^{n=k} \Delta I_n \Big/ \sum_{n=1}^{n=256} C_n \right) \times \frac{1}{6} = \frac{\Delta I \times k}{256 \times C} \times \frac{1}{6} \tag{3.20}$$

where k is the number of the enabled tri-state inverter cells. (3.20) is under the assumption that all the unit cells have the same loading capacitance (C) and the same driving current (ΔI). The DCO frequency is determined by the sum of loading capacitances (256 x C) and the sum of the unit currents ($\Delta I \times k$). In this simplified equation, the oscillation frequency should be exactly linear for$k$, but the simulation results for post-netlist exhibits a small distortion. In order to explain nonlinearity, the equation (3.20) should be modified as the following.

$$F_{DCO} = \frac{\Delta I \times k \times 1/6}{(C_{on} + C_{rc}) \times k + (C_{off} + C_{rc}) \times (256 - k)} \tag{3.21}$$

(3.21) shows the loading capacitance variation according to the on/off status of the unit tri-state buffer. Therefore, the DCO gain curve is not exactly linear for k because the loading capacitance has voltage dependency. That is, C_{on} is not equal to C_{off} . The effect of capacitance nonlinearity is decreased as the parasitic component C_{rc} is increased.

Another factor that degrades the linearity is the parasitic resistance in metal routing. A narrow metal routing with 28nm CMOS technology is used to save area and reduce loading capacitance at the cost of increased parasitic resistance. The parasitic resistance causes the nonlinearity in the DCO tuning curve, because the effective driving strength of each delay cell depends on the location where the transistors are turned on and off. In order to analyze the resistance effect, the metal routing is modeled using the equivalent π model as shown in Fig. 3.48. The resistance R_v of a vertical metal line degrades linearity by changing the effective driving strength of the connected delay cells. On the other hand, the resistance R_h of horizontal metal degrades only speed, because it only affects a horizontally connected single delay cell. The analytical model of (3.21) does not include the resistance effect, since it is too complicated and much too dependent on the layout style. Fig. 3.49 shows the

DCO tuning curves using the proposed model shown in Fig. 3.48. The output of the model is compared against the simulation results with RC extracted netlist and measurements, which show less than 3% difference across all codes. There is a trade-off between linearity and power consumption, so in order to enhance the linearity, the parasitic resistance (R_v) must be reduced by increasing the width of the metal line that inevitably increases the parasitic loading capacitance (C_{rc}). With the increased capacitance, the driver transistors size must also be increased to oscillate at the same frequency, thereby increasing power consumption

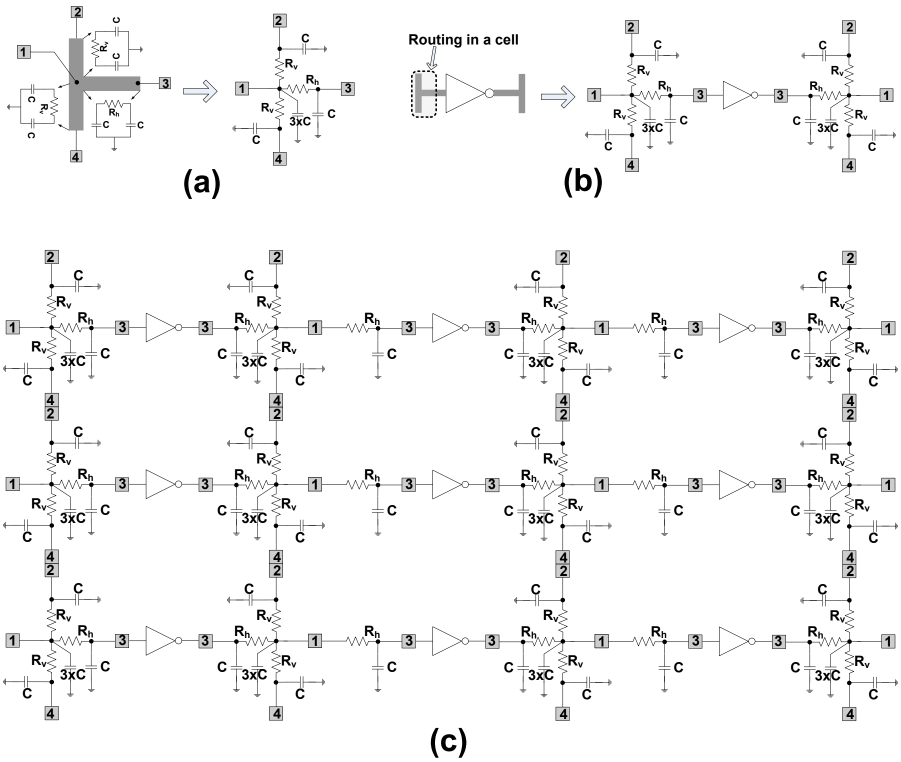


Fig. 3.48. The π model is used to model the internal routing path. (a) The π model for each routing path, (b) the equivalent circuit for the unit delay cell including the internal routing path, and (c) the equivalent circuit model for 3x3 DCO cells.

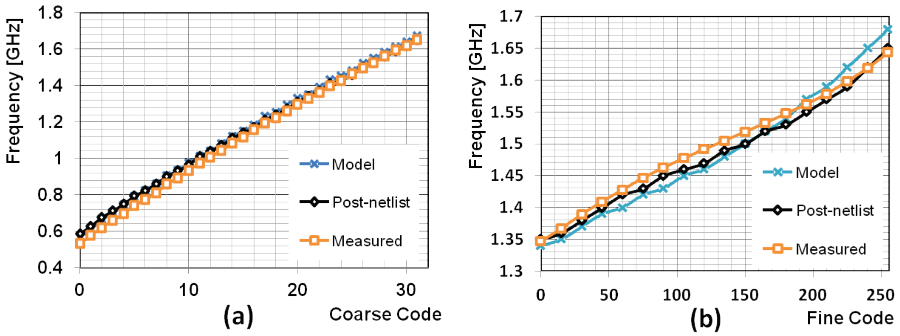


Fig. 3.49. DCO tuning curves for comparisons. (a) Coarse tuning and (b) fine tuning.

3.7 Experiment Results

The test chip is implemented in 28nm logic CMOS technology. Fig. 3.50 depicts the micrograph of the test chip.

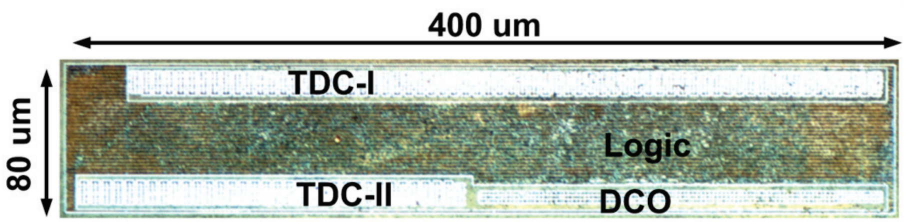


Fig. 3.50. Test chip photograph

3.7.1 DCO measurement

Fig. 3.49(a) shows the tuning curve for the 5-bit coarse tuning which ranges from 560 MHz to 1650 MHz with a 40-MHz resolution. The 8-bit fine control offers a 1-MHz resolution as shown in Fig. 3.49(b).

Fig. 3.51 shows that DNL ranges from -0.5 LSB to 1.7 LSBs and INL is less than ± 12 LSBs. In order to remove the effects of the random mismatches and measurement errors, the measured raw DNL data are mathematically post-processed using the moving average technique as depicted in Fig. 3.51(b). Since the DCO has a very fine resolution (~1 MHz), it is very difficult to measure the exact frequency, especially in a higher frequency band. The 16-point moving average is utilized to filter out such measurement errors, and the processed data shows only the systematic mismatch effect which is less than ± 0.3 LSB.

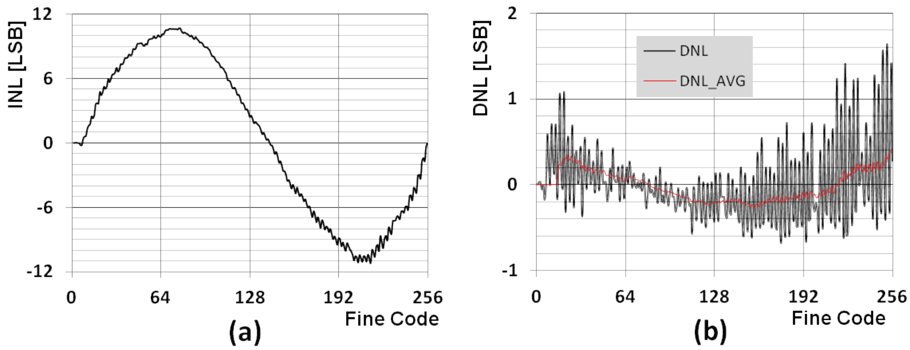


Fig. 3.51. DCO linearity characteristics. (a) INL and (b) raw DNL and 16-point averaged DNL.

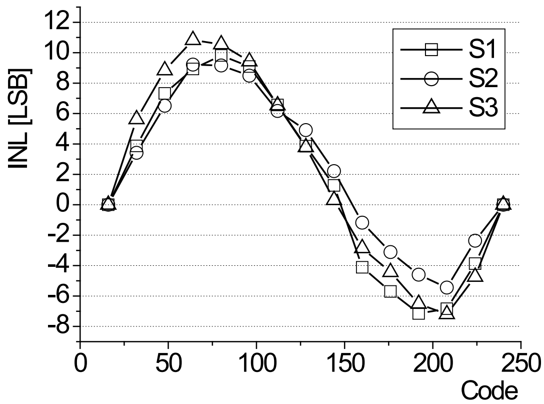


Fig. 3.52. Sample variations of INL.

Fig. 3.52 shows the variations over three sample chips that reveal the random device mismatch effect, where each curve has almost the same INL characteristics. This means the dominant linearity degradation factor is the systematic mismatch coming from the parasitic RC of the internal routing within unit cells. The linearity error can be compensated, because the degradation pattern is very deterministic. INL shows the third order curve and DNL has the second order shape.

Table. 3.8 summarizes the results comparing recently published ring oscillator DCOs and this work. The proposed DCO shows the finest resolution (0.37 ps) and the largest intrinsic tuning range (250 MHz ~ 1650 MHz). In particular, the linearity is comparable to the full-custom layout [71].

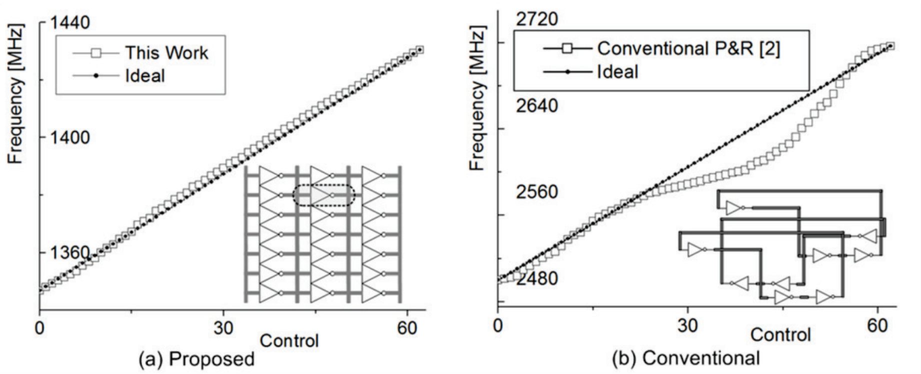


Fig. 3.53. Proposed Cell based Layout Vs Conventional Auto P&R

Table. 3.8. DCO Performance Comparisons

	This Work		Park [72] 2011 CICC	Nejad. [73] 2005 JSSC	Sheng. [70] 2007TCAS	Wu. [71] 2010TCAS
	(Fine)	(Coarse)				
Tuning method	Tristate Inv		Tristate Inv	DAC + CCO	Cap	Tristate Inv
Design method(1)	Synthesized		Synthesized	Custom	Custom	Custom
DNL [LSB]	-0.5~ 1.7	-0.3 ~ 0.3	N/A(3)	N/A	N/A	-0.95 ~ 1.2
INL[LSB]	-12 ~ 12	0 ~ 1.3	N/A(3)	N/A	N/A	N/A
Resolution [ps]	0.37ps	17ps	0.48ps	2ps	1.47ps	8.8ps
Tuning range [MHz]	250M ~ 1650M		1500M ~ 2700M	410M ~ 500M	191M ~ 952M	28M ~ 446M
Word length [bit]	8b	5b	8.3b	5b	15b	8b
Power [W] (2)	1.5mW @1500MHz		9mW @2.5GHz(2)	0.34mW @500MHz	0.14mW @200MHz	N/A
Power/Hz [W/GHz]	1.0 mW/GHz		3.6 mW/GHz	0.68 mW/GHz	0.7 mW/GHz	N/A
Voltage [V]	1.0 V		1.1 V	1.8 V	1.0 V	N/A
Process [nm]	28nm		65nm	180nm	90nm	180nm

(1) If there is no special comment for a synthesis then full-custom design is assumed.
(2) When only the PLL power is known, the DCO power is estimated to be 70% of the total PLL power.
(3) Park et. al does not provide the DCO tuning characteristics for a full 8-bit code
Gray colored areas represent the best performance among the compared ones.

3.7.2 **PLL measurement**

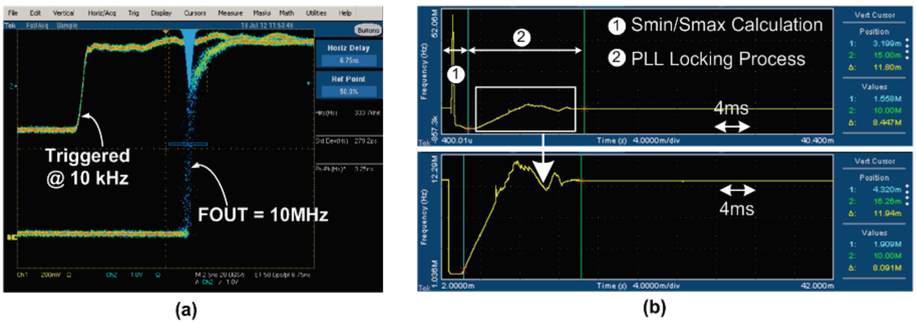


Fig. 3.54. Time domain measurement for HSYNC =10 kHz, FOUT = 10 MHz. (a) Phase tracking between HSYNC and output clock and (b) locking behavior.

Fig. 3.54(a) shows the tracking jitter between 10 kHz *HSYNC* and 10 MHz pixel clock. We measured at the lowest input frequency, because this is the worst condition possible for input phase tracking. The output clock jitter is measured with a trigger at the input clock and shows 3.25-ns p-p jitter, which is about 0.003% of the input period.

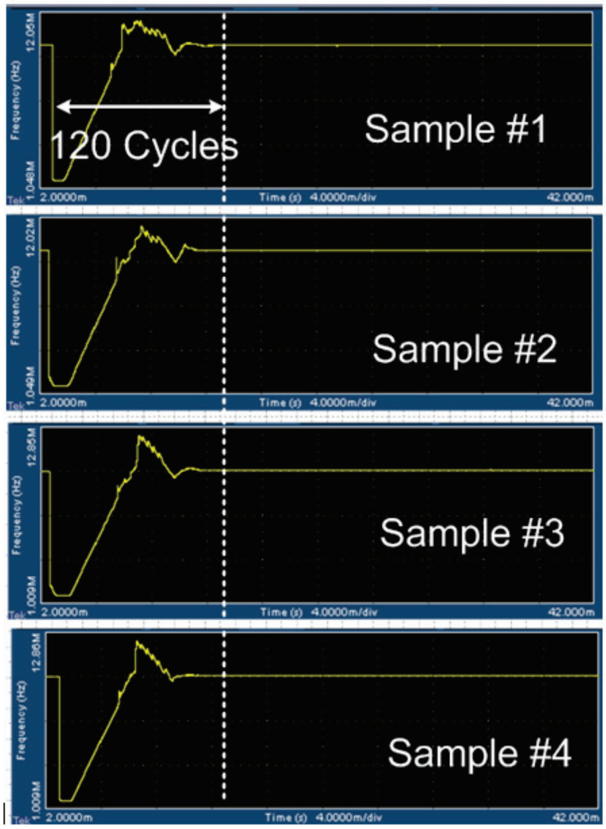


Fig. 3.55. Locking process measurements for different samples

The measured locking characteristic is depicted in Fig. 3.54(b), and is divided into 2 stages. The first step is the S_{MIN}/S_{MAX} calculation, where the available range of the S value is calculated using the circuit of Fig. 3.17, where the PLL operates in an open loop. After the S_{MIN}/S_{MAX} calculation is completed, the PLL moves on to a closed-loop operation mode. In the initial stage of this mode, the S value is adjusted until the *coarse* code is within its min/max range. After the S value is fixed, the *coarse* code is controlled so that the *fine* code is within the min/max range. Finally, the *fine* code is adjusted so that the phase error is minimized and the PLL locks within 120 cycles of the input clock. The locking process has been repeatedly measured for different samples to show the robustness of the proposed bottom-up control algorithms. Fig. 3.55 shows that the PLL always locks within 120 input cycles regardless of sample variation.

Fig. 3.56 shows the phase noise plot and integrated jitter for a 250 MHz output clock regenerated from the 100 kHz input signal. In order to show the effectiveness of the dual-loop PLL, the loop bandwidth of the fast-loop is varied. A large amount of *1/f* noise appears for the 100 kHz loop bandwidth condition because the DCO noise is not filtered out sufficiently. As verified using the linear model (Fig. 3.25), in-band noise can be removed efficiently by increasing the loop bandwidth of the fast-loop. Fig. 3.56(a) shows that the phase noise is reduced by as much as 25 dB at the lower offset frequency by increasing the bandwidth of the fast-loop from 100 kHz to 1.5 MHz. Fig. 3.56(b) presents the noise plot for a further optimized design that has enhanced TDC resolution and optimized IIR filtering coefficients. Compared to our prior report [65] shown in Fig. 3.56(a), (b) shows how the RMS integrated jitter is reduced from 30 ps to 15 ps. These experimental results coincide with the s-domain analysis results of Fig. 3.23, Fig. 3.24, and Fig. 3.25.

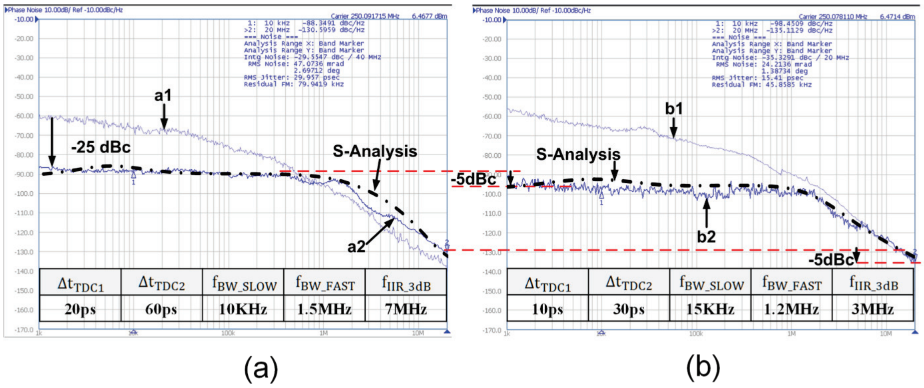


Fig. 3.56. Measured phase noise and integrated jitter for HSYNC =100 kHz, FOUT = 250 MHz. (a) Loop bandwidth optimization only [1] and (b) loop BW optimization + IIR filtering + TDC resolution enhancement. s-domain analysis is well matched to the measurements results. The inserted tables show the loop parameters for a2 and b2 respectively. The parameters of a1 and b1 are for the low loop bandwidth of 100 kHz in the fast-loop PLL.

Table. 3.9. Pixel Clock Generator Performance Comparisons

	This Work	Marie [59] 1998 JSSC	CHUNG [58] 2011 JSSC	Xiu [47] 2004 JSSC	Lee [63] 2006 ISSCC
Type	DUAL LOOP ADPLL	CHARGE PUMP PLL	SINGLE LOOP ADPLL	FLYING ADDER PLL	DUAL LOOP HYBRID PLL
Leakage problem	NO	SEVERE	NO	MINOR	MINOR
External Filter	NO	NECESSARY	NO	NO	NO
Design method (1)	SYNTHESIS	FULL CUSTOM	CUSTOM + SYNTHESIS	CUSTOM + SYNTHESIS	CUSTOM + SYNTHESIS
Process	28nm	1000nm	65NM	600nm	180nm
Supply Voltage	1.0 V	5 V	1.2 V	3.3 V	1.8 V
Power	3.1mW @250MHz	N/A	0.8MW @190MHz	180MW @200MHz	5mW @170MHz(4)
Size	0.032mm^2	N/A	0.07MM^2	1.8MM^2	0.23mm^2
Integrated Jitter	15ps$_{rms}$ @250MHz	250ps$_{rms}$ @80MHz(2)	210ps$_{rms}$ @190MHz (2)	190ps$_{rms}$ @210MHz (2)	21ps$_{rms}$ @190MHz
FOM (3)	1.4	N/A	12	61560	241

(1) If there is no special comment for a synthesis then it is dealt with the custom design.
(2) For the fair comparisons, RMS jitter values are compared. If RMS is not known then peak-peak value is divided by 8 to obtain RMS value.
(3) FOM is defined by the "FOM = Power * Size * Jitter". The smaller FOM means a better design.
(4) DSM is implemented with external FPGA chip. And the power consumption due to DSM is not included.

Gray colored areas denote the best performance among the compared works.

Table. 3.9 summarizes the comparison results for the previously reported pixel clock generators. The proposed dual-loop architecture shows a superior jitter performance to the single-loop PLLs [59] and [58], because the dual-loop architecture inherently reduces the intrinsic phase noise of the DCO. Compared with Lee's hybrid PLL [63], this work achieved less integrated jitter while consuming only 6% of power consumption and occupying only 14 % of the chip area; moreover, its layout has been automatically synthesized as shown in Fig. 3.50. For fair comparison, the performance numbers are compared based on the FOM, which is defined by the *"Size*Power*Jitter"*. This work shows the best FOM among the previously reported pixel clock generators.

3.8 Conclusions

The design of the conventional analog-digital hybrid PLL [63] has been converted to an all-digital scheme. In order to properly utilize the dual-loop architecture in a pure digital domain, a new bottom-up DCO control algorithm has been proposed. In addition, the s-domain noise analysis and the RC equivalent circuit model are utilized to obtain design insights and optimize loop parameters. The prototype chip has been synthesized using the proposed plug-in unit cells without performance degradation. The fabricated chip shows the lowest FOM having lowest RMS integrated jitter (15 ps), compact area (0.032mm^2), and low power consumption (3.1 mW at 1.0 V).

A. Device Technology Scaling Trends

In this chapter, we'll briefly overview the motivation and theory of the device scaling and introduce the design challenges in scaled technology [1]

A.1. Motivation for Technology Scaling

The definition of the device technology scaling is to reduce transistor geometry. As the device size shrinks, we can get two advantages.

The first, the more devices can be integrated within the same chip area. In a simple, if the device size shrinks by 1/k then the area consumption is reduced by $1/k^2$. This has been the main driving force of continuing the technology scaling. Fig.A.1 shows the gate length scaling roadmap which is provided by the *International Technology Roadmap for Semiconductors (ITRS)* [1]. From Fig.A.1, we can predict that the feature size becomes a half within 10 years; the integration density is approximately quadrupled.

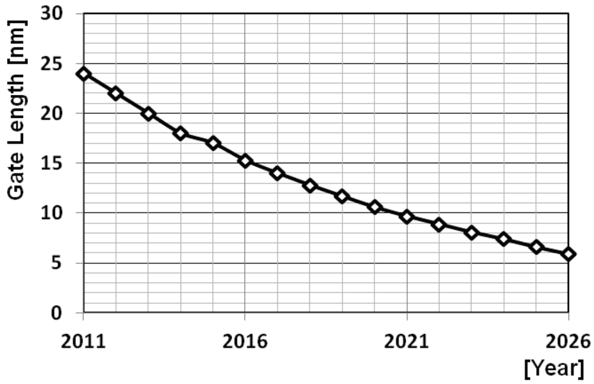


Fig.A.1. Si-MOSFET gate length scaling roadmap (ITRS 2012)

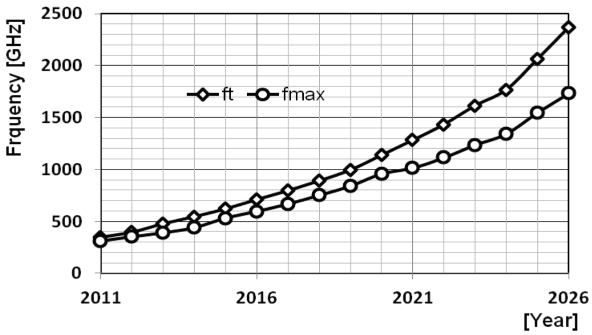


Fig.A.2. Si-MOSFET speed roadmap (ITRS 2012)

The second advantage is that the operation speed is improved because the electrons and holes travel across the channel in a shorter time as the channel length is reduced. And the smaller feature size also reduces a gate-source capacitance. Therefore it is easier to alternate the on/off states of a transistor and reduces the transit time of logic circuit; the logic circuit is basically on/off operation circuit. In the aspect of an analog circuit, the cut-off frequency f_T and self oscillation frequency f_{MAX} are used to represent a transistor speed. Fig.A.2 illustrates the speed trend with years. It shows that the Si-MOSFET technology approaches to THz (10^{12} Hz).

A.2. Constant Field Scaling

The principal scaling rule is to reduce the geometry of a transistor with keeping the transistor working properly. At a first glance, it seems that the transistor would operate well even if one shrinks the size without changing other parameters such as doping constraint, oxide thickness and bias voltage.

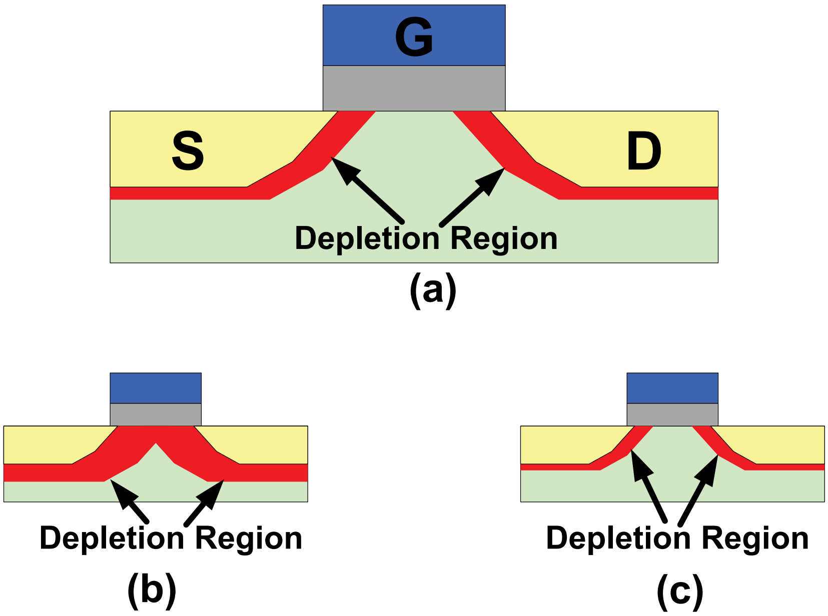


Fig.A.3. Basic convention of the transistor scaling. (a) Before scaling, (b) a source/drain punchthough due to the wrong scaling strategy without changing other parameter except geometry, (c) a proper constant electric field scaling.

Fig.A.3 shows that shrinking a size is not a simple matter. If we want to scale down an original size by a half, then we should also scale the other parameters such as doping density, oxide thickness, and bias voltage to operate the transistor properly. Fig.A.3 (b) shows that the depletion region is shorted together because the doping constraint is not properly scaled up. To guarantee a proper transistor operation, we have to scale the doping density too, as illustrated in Fig.A.3 (c). This device scaling strategy is called the *constant field scaling* because the electric field intensity is kept the same even after shrinking the geometry.

Table.A.1. Constant-field scaling

Quantity	Scaling factor
Device Dimensions (*L, W*)	1/k
Gate oxide thickness, d_{ox}	1/k
P-N junction depth, d_{junc}	1/k
Area per unit transistor	$1/k^2$
Devices per unit area	k^2
Doping Concentration, N_A	k
Bias Voltage and current	1/k
Threshold Voltage , V_{th}	1/k
Power dissipation for a given circuit	$1/k^2$
Power dissipation per unit of chip area	1
Capacitance	1/k
Capacitance per unit area	k
Electric field intensity	1
Body effect coefficient, γ	$1/k^{0.5}$
Transistor transit time, τ	1/k
Transistor power-delay product	$1/k^3$

Table.A.1 summarizes the constant electric field scaling [17]. The first step of the scaling is to reduce the gate length (*L*) and width (*W*) by 1/k factor. To keep the electric field as the same, the bias voltage and current should be also scaled down by 1/k. The junction depth should be scaled down by 1/k to prevent two depletion regions of source and drain from being shorted.

(A. 1) shows that the junction depth (d_{junc}) is inversely proportional to the doping concentration (N_A). V_{junc} is the applied junction voltage, Φ_0 is built-in potential of a junction, ε_s is the permittivity of Si substrate (0.104 fF/um). To scale the d_{junc} by 1/k, the N_A is increased by k and the V_{Junc} is decreased by 1/k; that is, the bias voltage is decreased by 1/k. Increasing the doping concentration will also increase the threshold voltage, this can be corrected by decreasing the oxide thickness by 1/k. Now the scaling is completed.

$$d_{junc} = \sqrt{\frac{2 \times \varepsilon_s \times (\Phi_0 + V_{Junc})}{q \times N_A}} \tag{A. 1}$$

The power consumption is also reduced by $1/k^2$ because the bias current and voltage are scaled by 1/k. But the power density per a unit area is not scaled because the number of devices within a unit area becomes k^2. This limits the integration level and operating frequency because the high operating frequency and large transistor packing density in a scaled technology will highly increase the power consumption of a chip.

The capacitance is scaled down by 1/k because the area is scaled by $1/k^2$ and the distance between electrode is reduced by 1/k; whole effect is calculated by $(1/k^2)\cdot 1/(1/k)$ and denoted by 1/k. However the capacitance per unit area is increased by k because the packing density is scaled up by k^2; whole effect is calculated by $(1/k) \cdot k^2$ and is equal to k.

(A. 2) shows that the transit time($\Delta\tau$) to charge and discharge a capacitor is proportional to the capacitance value (C) and voltage swing ($\triangle V$), but is inversely proportional to the charging current (I). The transit time will be scaled by 1/k because the net effect is calculated by (1/k)·(1/k)/(1/k) and equal to 1/k.

$$\Delta\tau = \frac{C \times \Delta V}{I} \tag{A. 2}$$

Previously, we have found the power dissipation per transistor is reduced by $1/k^2$. Therefore the power delay product (power x delay) is equal to $1/k^3$; it is calculated by $(1/k^2)\cdot(1/k)$ and is equal to $1/k^3$.

A.3. Quasi Constant Voltage Scaling

In a *constant field scaling* strategy, the supply voltage is scaled down by 1/k to guarantee a constant electric field intensity and to prevent breakdown failure. However, reducing a supply voltage is not always available in the real world. The first reason is that the sub-threshold slope is difficult to scale. In other words, the voltage swing of the gate should be large enough to turn off the device completely. And the second, lower supply voltage reduces a noise margin, and device becomes more susceptible to the V_{th} fluctuation caused by PVT variations.

The *constant voltage scaling* is proposed to solve this issue. In this scaling method, the *W, L,* and N_A are scaled by 1/k. But the supply voltage is kept constant. The oxide thickness is not scaled by 1/k but is scaled by 1/b; the b is less than k to prevent an oxide breakdown. The *constant voltage scaling* causes many harmful effects due to the exceedingly large electric filed.

In order to solve the problems of two extreme cases of constant voltage and current scaling, the *Quasi-constant voltage scaling* is generally adopted. In this scheme, the supply voltage is scaled by b which is less than k. More generalized scaling strategy can be also used considering a target performance. In this case, the doping concentration, supply voltage and threshold voltage are optimized to meet a target performance. Table.A.2 summarizes the various scaling method.

Table.A.2. Scaling Rules. ($1 < b < k$)

Quantity	Constant electric field scaling	Constant voltage scaling	Quasi-constant Voltage scaling	Generalized scaling
W, L	1/k			
d_{ox}	1/k	1/b	1/k	1/k
N_A	k	k	k	k^2/b
VDD, V_{th}	1/k	1	1/b	1/b

A.4. Device Technology Trends in Real World

In this section, the ITRS reports are summarized in terms of a circuit design. The ITRS report shows that a device size is scaled by 0.7 times as a technology node moves onto next one [1]. The relation between adjacent technologies nodes is expressed by (A. 3), where L_{next} is the minimum feature size of the following next technology node, and $L_{present}$ is the one for the present technology node. Using this equation, we can forecast the next technology node. Fig.A.4 illustrates the calculated trend and real trend are matched well.

$$L_{next} = L_{present} \times 0.7 \tag{A. 3}$$

As shown in Table.A.2, Fig.A.5, Fig.A.6, and Fig.A.7, the oxide thickness should be scaled down to adjust a threshold voltage. The thickness is slightly different according to the device architectures. Generally speaking, the muti-gate architecture such as FinFET has the larger thickness compared to the single gate transistor architectures (Fig.A.5).

Fig.A.6 presents the supply voltage and threshold voltage trends for the various Si-MOSFET technologies. While the supply voltage is continuously scaled down, but the threshold voltage is not changed a lot. It means the over drive voltage (Vgs-Vth) decreases as the scaling proceeds. This can be a issue in an analog circuit, because a cascode scheme cannot be used due to small voltage head room.

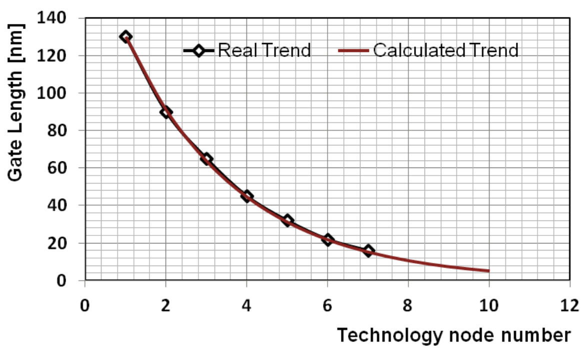


Fig.A.4. Technology Trend Forecast. (ITRS 2012)

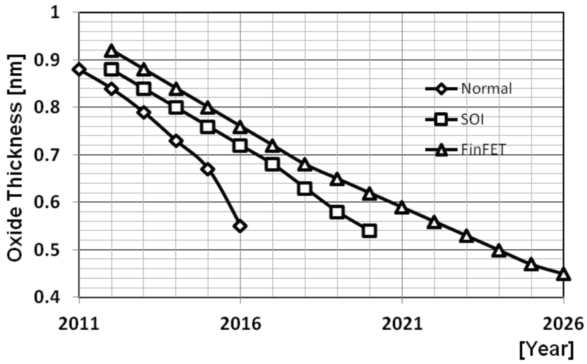


Fig.A.5. Oxide Thickness Trends. (ITRS 2012)

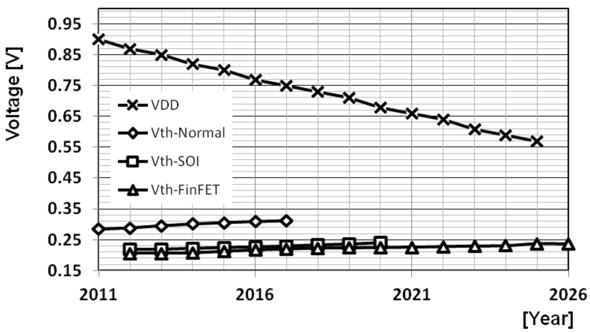


Fig.A.6. Supply Voltage and Threshold Voltage Trend. (ITRS 2012)

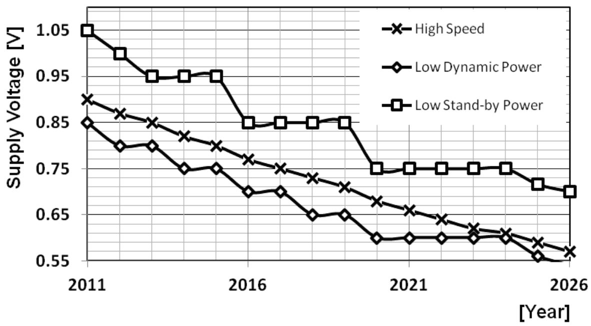


Fig.A.7. Supply Voltage Trends for Different Transistor Options. (ITRS 2012)

In a modern advance technology, it provides multiple transistor technology options having different oxide thickness. A user chooses a proper device type according to an application. A high speed transistor has large current driving.

A low dynamic power option is focused on a reduction in dynamic switching power. Finally, a low stand-by power technology is designed to reduce the static current during off state. The driving current is expected to increase continuously for a higher speed (Fig.A.8). And the device architectures are forecasted to evolve from a normal planar type to a multi-gated architecture (FinFET). According to the transistor options, the current density is a little different as shown in Fig.A.9.

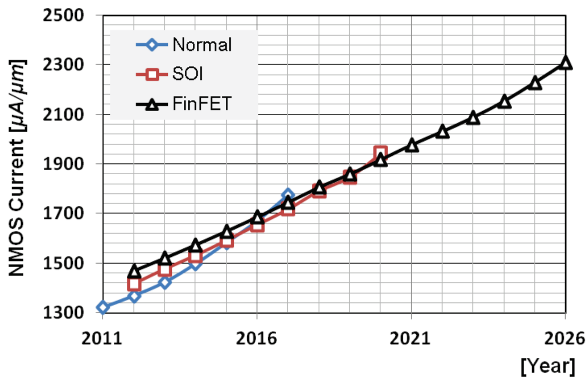


Fig.A.8. NMOS Current per Gate Width. (ITRS 2012)

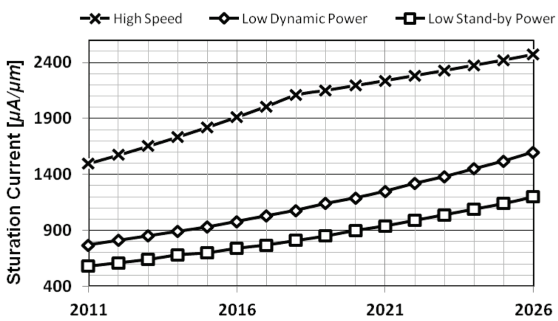


Fig.A.9. Saturation Current Trends for Different Transistor Options. (ITRS 2012)

Fig.A.10 presents the off state drain/source current comparisons between transistor options; note that y-axis is log scale. The low stand-by transistor has a negligible off-state current sacrificing the current driving capability. Otherwise, the high speed option has a large off-state leakage current. The low dynamic option is placed between two extremes; it has acceptable dynamic current and off-state current.

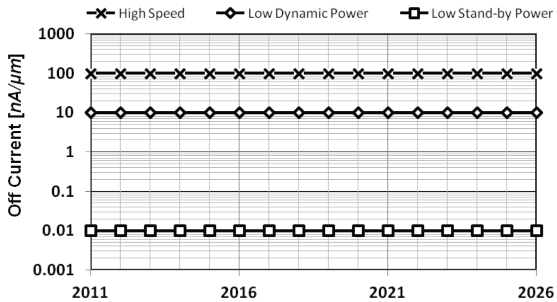


Fig.A.10. Off-State Current Trends for Different Transistor Options.
(ITRS 2012)

According to the constant field scaling theory, a gate capacitance is scaled down by 1/k as the device size shrinks (Table.A.1). And Fig.A.11 depicts the capacitance trend is coincident with the theory. The interesting thing is the fringing capacitance portion is not scaled down significantly; therefore fringing capacitance will become a more dominant component among total capacitance, which is illustrated in Fig.A.12.

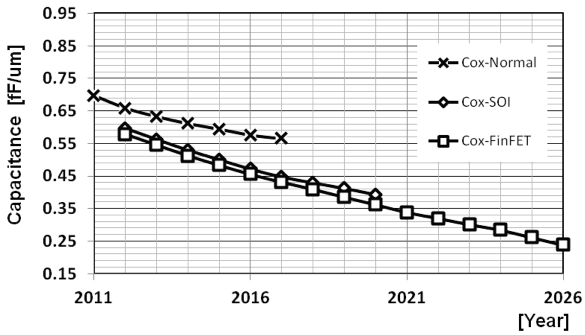


Fig.A.11. Gate Capacitance per Width. (ITRS 2012)

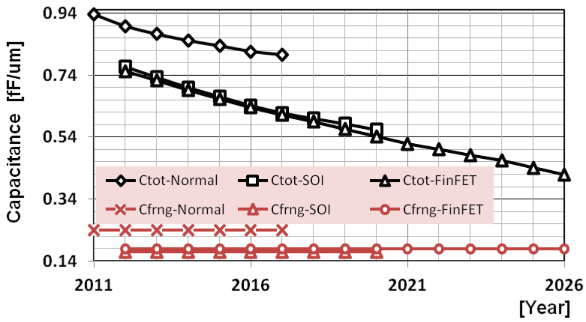


Fig.A.12. Total Capacitance per Width, and Fringing Capacitance per Width. (ITRS 2012)

A reduction in total capacitance contributes to the reduction in a dynamic power and speed improvement, which are shown in Fig.A.13 and Fig.A.14. The improved speed can be measured by the unit timing delay of ring oscillator (Fig.A.15). In the aspect of analog circuit, frequency domain index is more useful to determine the high frequency performance. Fig.A.16 and Fig.A.17 provide the cut-off frequency and maximum oscillation frequency trends respectively.

Table.A.3 compares the silicon MOS technologies and compound semiconductor technology in terms of power, speed, and static power.

Compared to the high speed silicon MOS, the III/IV technology is 1.5x faster, and consumes less power.

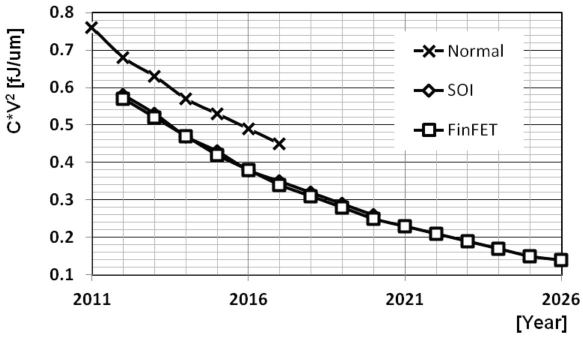


Fig.A.13. Dynamic Power Indicator per Width. (ITRS 2012)

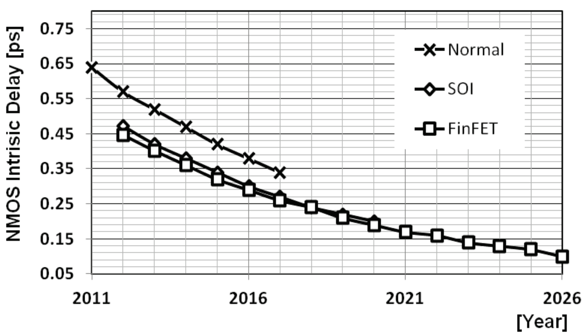


Fig.A.14. NMOSFET Intrinsic Delay. (ITRS 2012)

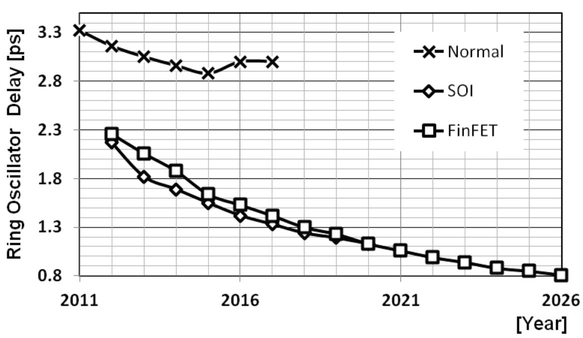


Fig.A.15. Ring Oscillator Delay per Unit Stage. (ITRS 2012)

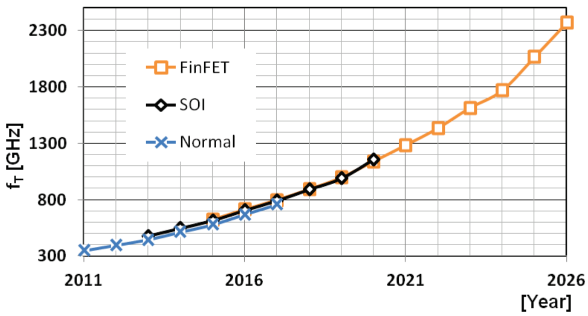


Fig.A.16. Cut-off Frequency (f_T). (ITRS 2012)

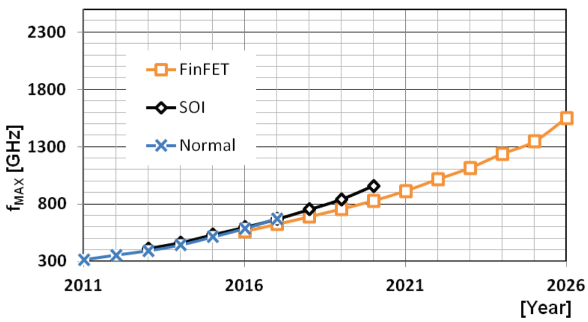


Fig.A.17. Maximum Oscillation Frequency Trend. (ITRS 2012)

Table.A.3. Comparison of Transistor Technologies. (ITRS 2012)

Transistor Type / Performance	Silicon MOSFET Technology			III-V/Ge Technology
	High Speed	Low Dynamic Power	Low Stand-by Power	
Speed (I/CV)	1	0.5	0.25	**1.5**
Dynamic Power (CV^2)	1	**0.6**	1	**0.6**
Static Power (I_{off})	1	0.05	**0.0001**	1

Fig.A.18 provides an intrinsic voltage gain trends. ITRS forecasts it will decrease continuously due to short channel effects. The degradation in intrinsic gain makes design issues in high performance amplifier design. Thus, various researches have been done to overcome this challenge [2].

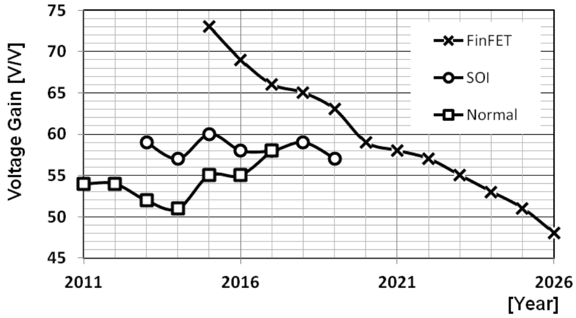


Fig.A.18. Analog Transistor Voltage Gain @ 10% $I_{D,sat}$ and 5x Minimum Gate Length. (ITRS 2012)

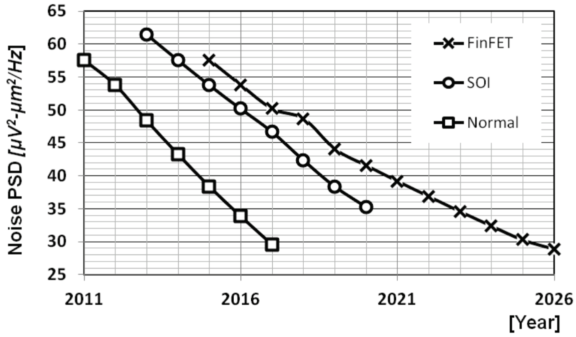


Fig.A.19. *1/f* Noise Power Spectral Density. (ITRS 2012)

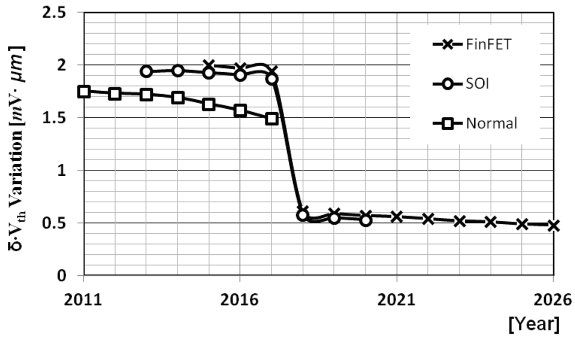


Fig.A.20. V_{th} Varaion per Unit Distance. (ITRS 2012)

Fig.A.19 shows the flicker noise performance requirements in the future. The more stringent noise performance will be required due to more advanced system specifications. Fig.A.20 presents the matching characteristics of threshold voltage. As the device fabrication technology is advanced, the matching characteristic is expected to improve at near 2017; which is forecasted based on the roadmap for device architecture and material technology.

Until now, we have overviewed the forecasted trends for active device. Now we'll show the trends for passive devices.

Fig.A.21, Fig.A.22, Fig.A.23, and Fig.A.24 are the trends for on-chip resistors. Fig.A.21 presents the sheet resistance for metal and poly resistor. The value is not scaled down because it's basically determined by the used material. The poly resistor has 5x larger sheet resistance. In the aspect of matching, the metal resistor has currently is better, but the poly will achieve a comparable matching characteristics due to a more advanced process technology (Fig.A.22). Both resistors have the same temperature coefficient (Fig.A.23). but the poly resistor has larger parasitic capacitance than metal one as shown in (Fig.A.24).

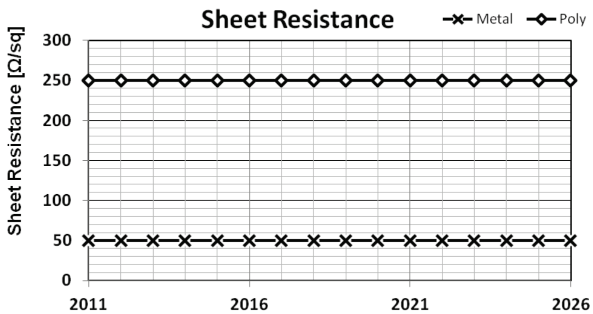


Fig.A.21. Sheet Resistance Trends for On-Chip Resistors. (ITRS 2012)

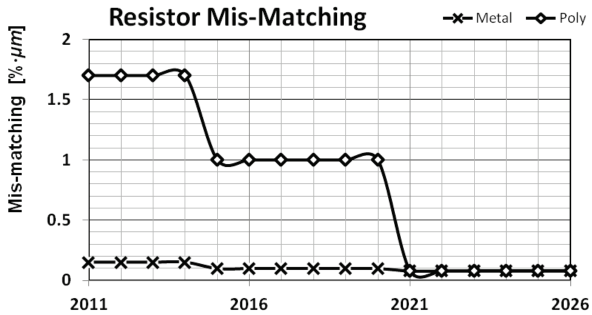


Fig.A.22. On-chip Resistor Mis-match Characteristics Trend. (ITRS 2012)

The trends for on-chip capacitance are depicted in Fig.A.25 ~ Fig.A 28. Unlike a resistance, a capacitance depends on both material and geometry. Fig.A.25 shows that the capacitance density increases as the electrodes space is scaled down. Especially, the inter-metal capacitor overtakes the MOS capacitor and MiM one. This is because it's relatively easier to draw two metal line (Inter-metal cap) closely rater than to grow a thin dielectric film (MOS, MiM). In addition, the thin dielectric thickness increases a leakage current as shown in Fig.A.26. The MiM capacitor has less leakage current due to thicker dielectric layer; MiM does not use the gate oxide. In the aspect of matching characteristics, MOS capacitor is more difficult to achieve good uniformity because it has a shallow oxide thickness (Fig.A.27).

Finally, Fig.A 28 presents the expected quality factors of varactor and inductor to meet future system specifications. Generally speaking, inductor determines the whole quality factor of a system.

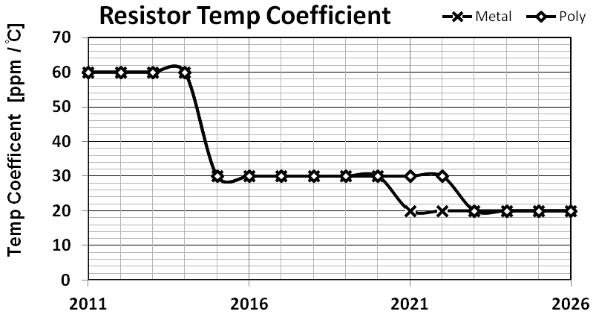


Fig.A.23. On-chip Resistor Temperature Coefficient Trend. (ITRS 2012)

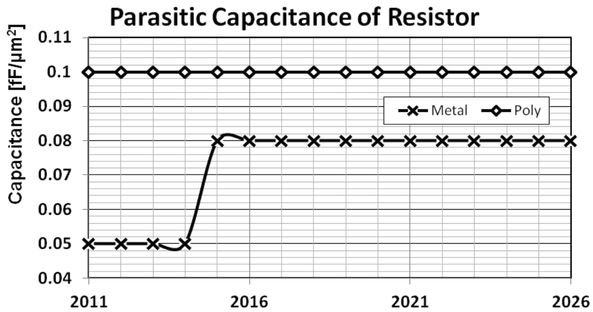


Fig.A.24. The Parasitic Capacitance of On-chip Resistor. (ITRS 2012)

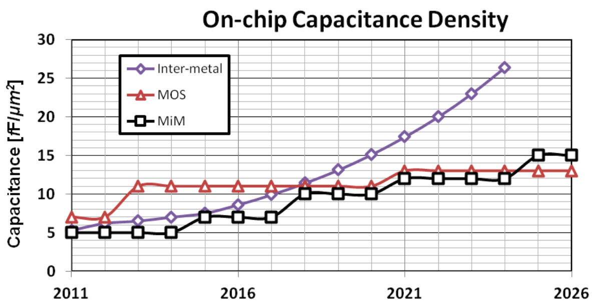


Fig.A.25. On-Chip Capacitance Density. (ITRS 2012)

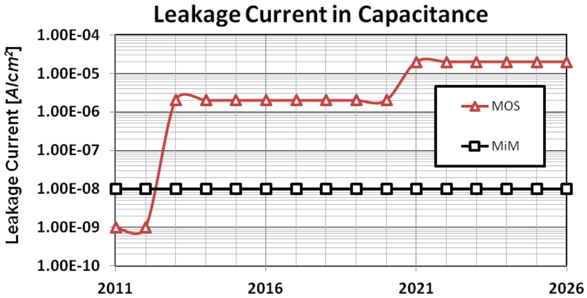


Fig.A.26. Leakage Current in On-Chip Capacitance. (ITRS 2012)

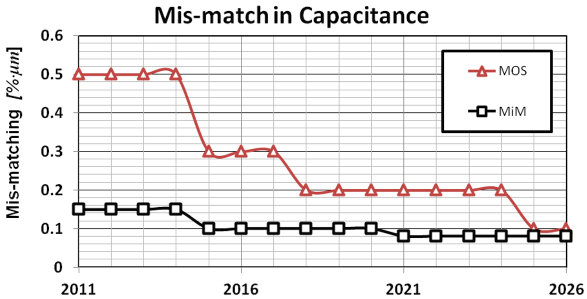


Fig.A.27. Mis-matching of On-Chip Capacitance. (ITRS 2012)

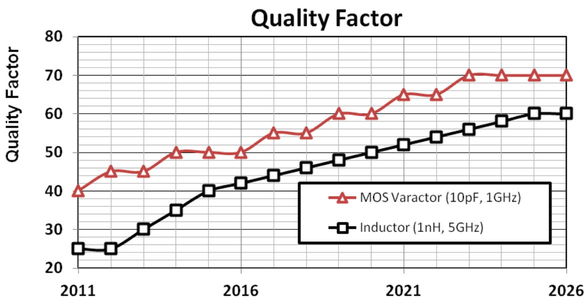


Fig.A 28. Quality Factor of Inductor and MOS Varactor. (ITRS 2012)

B. Spice Simulation Tip for a DCO

The DCO unit cell should be designed manually and Spice simulation is required to determine the number of unit-cells and find an exact tuning characteristics. Fig.B.1 illustrates the control methods of VCO and DCO. When one sweeps the input of DCO, all of the DCO control bits should be sequentially swept to generate a monotonically changing input digital code.

However, this approach increases a simulation time significantly and it's very tedious job to enter the entire control bit. To complete a DCO simulation simply, an ideal DAC is inserted as shown in Fig.B.2. The DAC is described in the Spice statement to execute the simulation under Spice environment, as shown in Fig.B.3. The analog input value of the DAC is defined by "vcon" parameter. Thus, the DAC can be conveniently swept by change the "vcon" parameter as the following transient simulation statement.

.tran 0.1n 10n sweep vcon 0 pdd 'pdd/256'

If a DCO has a thermometer input then DAC can be modeled as shown in Fig.B. 4. The ‘pdd’ of Fig.B.3 and Fig.B. 4 denotes a supply voltage value.

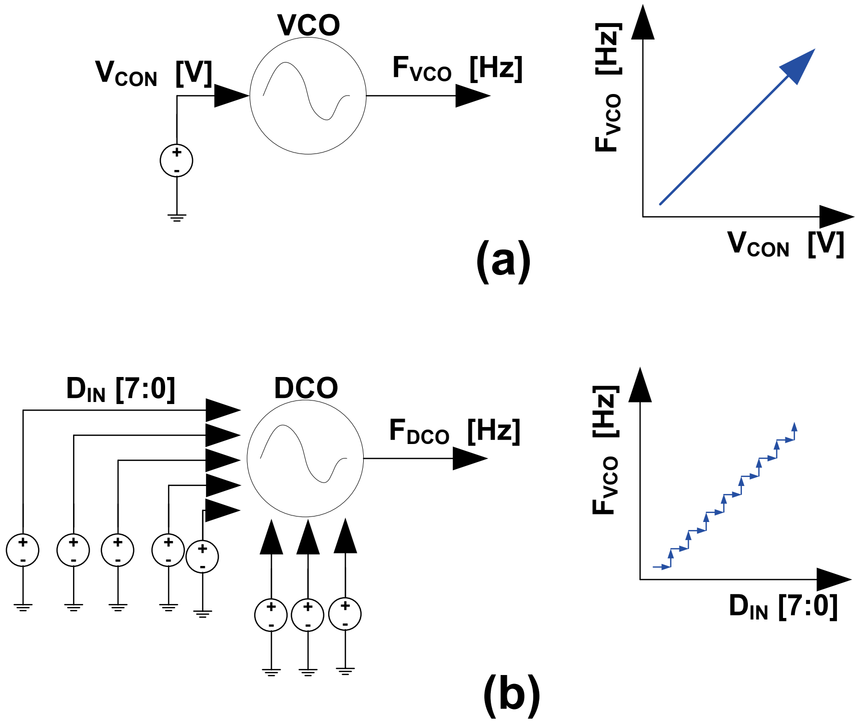


Fig.B.1. Input Control of VCO and DCO

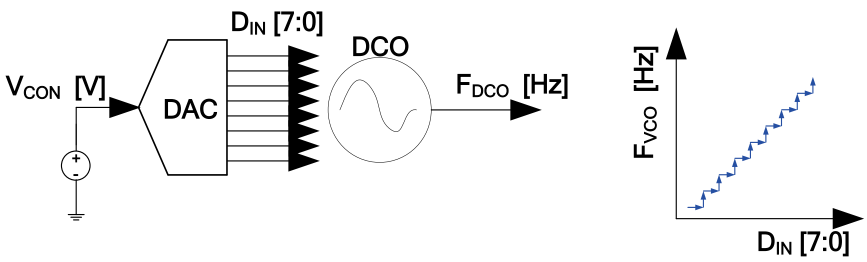


Fig.B.2. DCO simulation using an ideal DAC described in Spice.

```
.param
+ vf1_m_0=int(vcon/2)
+ vf1_m_1=int(vf1_m_0/2)
+ vf1_m_2=int(vf1_m_1/2)
+ vf1_m_3=int(vf1_m_2/2)
+ vf1_m_4=int(vf1_m_3/2)
+ vf1_m_5=int(vf1_m_4/2)
+ vf1_m_6=int(vf1_m_5/2)
+ vf1_m_7=int(vf1_m_6/2)
+ vf1_m_8=int(vf1_m_7/2)

+ vf1_r_0='int(vf1_val) -(2*int(vf1_val/2))'
+ vf1_r_1='vf1_m_0 -(2*int(vf1_m_0/2))'
+ vf1_r_2='vf1_m_1 -(2*int(vf1_m_1/2))'
+ vf1_r_3='vf1_m_2 -(2*int(vf1_m_2/2))'
+ vf1_r_4='vf1_m_3 -(2*int(vf1_m_3/2))'
+ vf1_r_5='vf1_m_4 -(2*int(vf1_m_4/2))'
+ vf1_r_6='vf1_m_5 -(2*int(vf1_m_5/2))'
+ vf1_r_7='vf1_m_6 -(2*int(vf1_m_6/2))'
+ vf1_r_8='vf1_m_7 -(2*int(vf1_m_7/2))'

vf1_8 vf1_8 gnd 'vf1_r_8 *pdd'
vf1_7 vf1_7 gnd 'vf1_r_7 *pdd'
vf1_6 vf1_6 gnd 'vf1_r_6 *pdd'
vf1_5 vf1_5 gnd 'vf1_r_5 *pdd'
vf1_4 vf1_4 gnd 'vf1_r_4 *pdd'
vf1_3 vf1_3 gnd 'vf1_r_3 *pdd'
vf1_2 vf1_2 gnd 'vf1_r_2 *pdd'
vf1_1 vf1_1 gnd 'vf1_r_1 *pdd'
vf1_0 vf1_0 gnd 'vf1_r_0 *pdd'
```

Fig.B.3. Ideal DAC modeling using Spice. Digital output is binary code.

vcol_255 col_255 gnd 'col255*pdd'	.param col255=colval>255
.	.
...	...
omitted	omitted
...	...
.	.
vcol_9 col_9 gnd 'col9*pdd'	.param col9=colval>9
vcol_8 col_8 gnd 'col8*pdd'	.param col8=colval>8
vcol_7 col_7 gnd 'col7*pdd'	.param col7=colval>7
vcol_6 col_6 gnd 'col6*pdd'	.param col6=colval>6
vcol_5 col_5 gnd 'col5*pdd'	.param col5=colval>5
vcol_4 col_4 gnd 'col4*pdd'	.param col4=colval>4
vcol_3 col_3 gnd 'col3*pdd'	.param col3=colval>3
vcol_2 col_2 gnd 'col2*pdd'	.param col2=colval>2
vcol_1 col_1 gnd 'col1*pdd'	.param col1=colval>1
vcol_0 col_0 gnd 'col0*pdd'	.param col0=colval>0

Fig.B. 4. Ideal DAC having thermometer output code.

C. Phase Noise to Jitter Conversion

A phase noise is converted to a RMS integrated jitter using (C.1) [74], where f_c is a center frequency, and $\mathcal{L}(f)$ is a phase noise in [dBc/Hz].

$$TJ_{RMS} = \frac{1}{2\pi f_c} \cdot \sqrt{2 \cdot \int 10^{\frac{\mathcal{L}(f)}{10}} df} \quad [sec] \tag{C.1}$$

Fig.C.1 illustrates a conventional noise profile of a PLL. Fig.C.1 (a) is the output noise due to a input referred noise like TDC and input clock. Fig.C.1 (b) denotes the DCO referred noise, where the blue colored unbroken line is the random phase noise of DCO and the red colored dotted line is the shaped PLL output noise. The total noise is sum of Fig.C.1 (a) and Fig.C.1 (b) and denoted as the blue colored broken line of the Fig.C.1 (c). To simplify an analysis, the blue colored curve can be piecewise linearly approximated as the red colored line. The in-band noise is assumed to be constant from DC to frequency “f1” and starts to fall off having -40 dBc/dec slope. When the frequency offset is larger than "f2", the phase noise keeps a constant value, which is determined by a minimum noise floor. Now, we can describe a noise profile using two points "A" and "B" because the roll-off slope is always -40 dBc/dec for 2nd-order system. The frequency "f_1" is approximately the same with a PLL loop bandwidth, and the thermal noise cut-off frequency "f_2" is determined by DCO characteristics. The RMS jitter integrated from f_0 to f_1 is calculated as

$$\mathrm{TJ_{f01}} = \frac{1}{2\pi f_c} \cdot \sqrt{2 \cdot \left(10^{\left(\frac{L1}{10}\right)} \cdot \left[f_0^{\left(\frac{L1}{10}\right)}\right] \cdot \left(\frac{0}{10}+1\right)^{-1} \cdot \left[f_1^{\left(\frac{0}{10}+1\right)} - f_0^{\left(\frac{0}{10}+1\right)}\right]\right)}$$

In a similar manner, the jitter for f1~f2 and f2~f3 regions are calculated as the followings.

$$\mathrm{TJ_{f12}} = \frac{1}{2\pi f_c} \cdot \sqrt{2 \cdot \left(10^{\left(\frac{L1}{10}\right)} \cdot \left[f_1^{\left(\frac{L1}{10}\right)}\right] \cdot \left(\frac{-40}{10}+1\right)^{-1} \cdot \left[f_2^{\left(\frac{-40}{10}+1\right)} - f_1^{\left(\frac{-40}{10}+1\right)}\right]\right)}$$

$$\mathrm{TJ_{f23}} = \frac{1}{2\pi f_c} \cdot \sqrt{2 \cdot \left(10^{\left(\frac{L2}{10}\right)} \cdot \left[f_2^{\left(\frac{L2}{10}\right)}\right] \cdot \left(\frac{0}{10}+1\right)^{-1} \cdot \left[f_3^{\left(\frac{0}{10}+1\right)} - f_2^{\left(\frac{0}{10}+1\right)}\right]\right)}$$

The jitter for an entire region is calculated as

$$\mathrm{TJ_{RMS}} = \sqrt{(\mathrm{TJ_{f01}})^2 + (\mathrm{TJ_{f12}})^2 + (\mathrm{TJ_{f23}})^2} \quad [\mathrm{sec}]$$

Fig.C.2 illustrates that Fig. 3.56 (b) is approximated as piecewise linear curve and its integrated jitter is calculated using the proposed equations. The RMS jitter integrated from 1 kHz to 1 GHz is calculated as 16.27ps, which is very close to the measured value (15.4ps) of Fig. 3.56 (b).

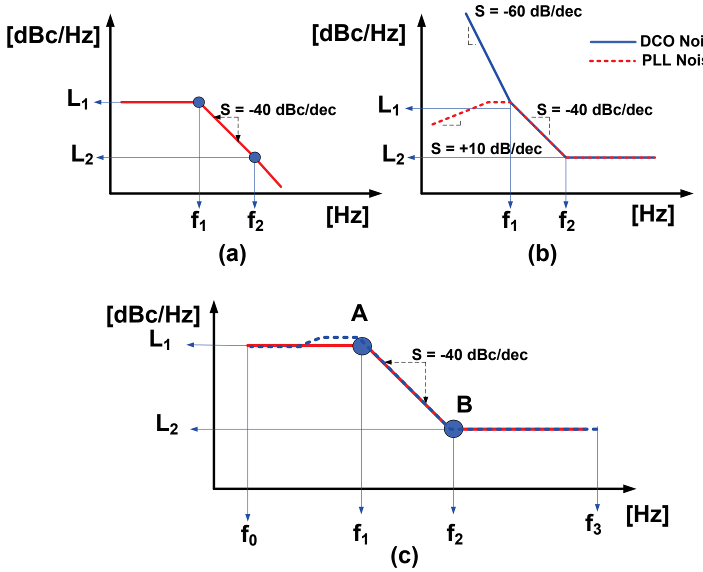


Fig.C.1. Conventional PLL Noise Profile. (a) Input path noise, (b) DCO path Noise, (c) Total Noise

$$TJ_{RMS} = \sqrt{(TJ_{f01})^2 + (TJ_{f12})^2 + (TJ_{f23})^2} = \sqrt{(13.87)^2 + (8.01)^2 + (2.82)^2} = 16.27\ [ps]$$

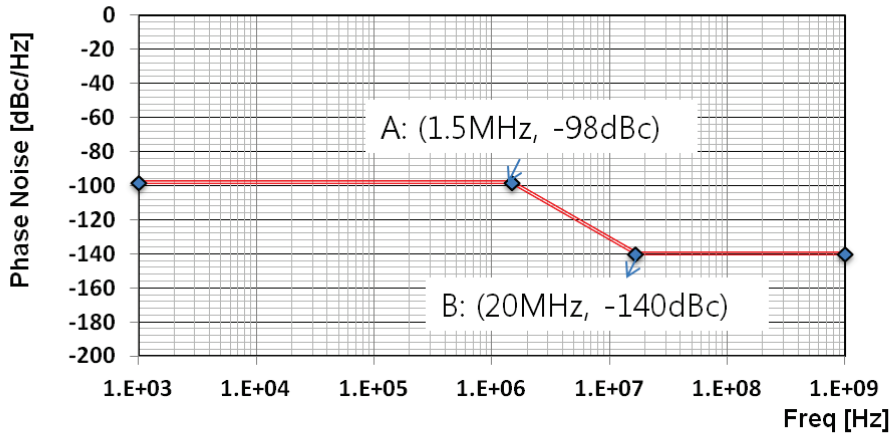


Fig.C.2. Phase noise to Jitter Conversion Example.

Bibliography

[1] http://www.itrs.net/Links/2012ITRS/Home2012.htm, "ITRS Report 2012 Edition," *International Technology Roadmap for Semiconductors (ITRS),* 2012.

[2] L. L. Lewyn, T. Ytterdal, C. Wulff, and K. Martin, "Analog circuit design in nanoscale CMOS technologies," *Proceedings of the IEEE,* vol. 97, pp. 1687-1714, 2009.

[3] A. Harjimiri, S. Limotyrakis, and T. Lee, "Jitter and phase noise in ring oscillator," *IEEE Journal of Solid-State Circuits,* vol. 34, pp. 790-804, 1999.

[4] R. Navid, T. H. Lee, and R. W. Dutton, "Minimum achievable phase noise of RC oscillators," *Solid-State Circuits, IEEE Journal of,* vol. 40, pp. 630-637, 2005.

[5] S. Desai, P. Trivedi, and V. Von Kanael, "A Dual-Supply 0.2-to-4GHz PLL Clock Multiplier in a 65nm Dual-Oxide CMOS Process," in *Solid-State Circuits Conference, 2007. ISSCC 2007. Digest of Technical Papers. IEEE International*, 2007, pp. 308-605.

[6] C.-H. Lee, K. McClellan, and J. Choma Jr, "Supply noise insensitive PLL design through PWL behavioral modeling and simulation," *Circuits and Systems II: Analog and Digital Signal Processing, IEEE Transactions on,* vol. 48, pp. 1137-1144, 2001.

[7] E. Alon, J. Kim, S. Pamarti, K. Chang, and M. Horowitz, "Replica compensated linear regulators for supply-regulated phase-locked loops," *Solid-State Circuits, IEEE Journal of,* vol. 41, pp. 413-424, 2006.

[8] A. Arakali, S. Gondi, and P. K. Hanumolu, "Low-power supply-regulation techniques for ring oscillators in phase-locked loops using a split-tuned architecture," *Solid-State Circuits, IEEE Journal of,* vol. 44, pp. 2169-2181, 2009.

[9] A. Elshazly, R. Inti, W. Yin, B. Young, and P. K. Hanumolu, "A 0.4-to-3 GHz Digital PLL With PVT Insensitive Supply Noise Cancellation Using Deterministic Background Calibration," *Solid-State Circuits, IEEE Journal of,* vol. 46, pp. 2759-2771, 2011.

[10] C.-H. Lee, K. McClellan, and J. Choma Jr, "A supply-noise-insensitive CMOS PLL with a voltage regulator using DC-DC capacitive converter," *Solid-State Circuits, IEEE Journal of,* vol. 36, pp. 1453-1463, 2001.

[11] T. Wu, K. Mayaram, and U.-K. Moon, "An on-chip calibration technique for reducing supply voltage sensitivity in ring oscillators," *Solid-State Circuits, IEEE Journal of,* vol. 42, pp. 775-783, 2007.

[12] Z.-X. Zhang, H. Du, and M. S. Lee, "A 360 MHz 3 V CMOS PLL with 1 V peak-to-peak power supply noise tolerance," in *Solid-State Circuits Conference, 1996. Digest of Technical Papers. 42nd ISSCC., 1996 IEEE International*, 1996, pp. 134-135, 431.

[13] V. Gupta, G. A. Rincn-Mora, and P. Raha, "Analysis and design of monolithic, high PSR, linear regulators for SoC applications," in *SOC Conference, 2004. Proceedings. IEEE International*, 2004, pp. 311-315.

[14] E. J. Pankratz and E. Sanchez-Sinencio, "Multiloop High-Power-Supply-Rejection Quadrature Ring Oscillator," *Solid-State Circuits, IEEE Journal of,* vol. 47, pp. 2033-2048, 2012.

[15] A. Arakali, S. Gondi, and P. K. Hanumolu, "Analysis and design techniques for supply-noise mitigation in phase-locked loops," *Circuits and Systems I: Regular Papers, IEEE Transactions on,* vol. 57, pp. 2880-2889, 2010.

[16] P.-H. Hsieh, J. Maxey, and C.-K. Yang, "Minimizing the supply sensitivity of a CMOS ring oscillator through jointly biasing the supply and control voltages," *Solid-State Circuits, IEEE Journal of,* vol. 44, pp. 2488-2495, 2009.

[17] Y. P. Tsividis, *Operation and Modeling of The MOS Transistors*, 1st ed.: McGraw-Hill, 1998.

[18] R. Holzer, "A 1 V CMOS PLL designed in high-leakage CMOS process operating at 10-700 MHz," in *Solid-State Circuits Conference, 2002. Digest of Technical Papers. ISSCC. 2002 IEEE International*, 2002, pp. 272-466.

[19] C.-C. Hung and S.-I. Liu, "A leakage-suppression technique for phase-locked systems in 65nm CMOS," in *Solid-State Circuits Conference-Digest of Technical Papers, 2009. ISSCC 2009. IEEE International*, 2009, pp. 400-401,401 a.

[20] 박홍준, *CMOS 디지털 집적회로 설계*: 홍릉과학출판사, 2008.

[21] J. V. Faricelli, "Layout-dependent proximity effects in deep nanoscale CMOS," in *Custom Integrated Circuits Conference (CICC), 2010 IEEE*, 2010, pp. 1-8.

[22] P. G. Drennan, M. L. Kniffin, and D. R. Locascio, "Implications of Proximity Effects for Analog Design," in *Custom Integrated Circuits Conference, 2006. CICC '06. IEEE*, 2006, pp. 169-176.

[23] K. Shu and E. Sanchez-Sinencio, *CMOS PLL synthesizers: analysis and design*: Springer Publishing Company, Incorporated, 2005.

[24] R. C. van de Beek, E. A. Klumperink, C. S. Vaucher, and B. Nauta, "Low-jitter clock multiplication: a comparison between PLLs and DLLs," *Circuits and Systems II: Analog and Digital Signal Processing, IEEE Transactions on,* vol. 49, pp. 555-566, 2002.

[25] T.-C. Lee and K.-J. Hsiao, "The design and analysis of a DLL-based frequency synthesizer for UWB application," *Solid-State Circuits, IEEE Journal of,* vol. 41, pp. 1245-1252, 2006.

[26] K.-H. Cheng, C.-W. Su, M.-J. Wu, and Y.-L. Chang, "A wide-range DLL-based clock generator with phase error calibration," in *Electronics, Circuits and Systems, 2008. ICECS 2008. 15th IEEE International Conference on*, 2008, pp. 798-801.

[27] F.-R. Liao and S.-S. Lu, "An injection-locked ring PLL with self-aligned injection window," *Microwave Theory and Techniques, IEEE Transactions on,* vol. 60, pp. 1086-1096, 2012.

[28] J. Begueret, Y. Deval, O. Mazouffre, A. Spataro, P. Fouillat, E. Benoit*, et al.*, "Clock generator using factorial DLL for video applications," in *Custom Integrated Circuits, 2001, IEEE Conference on.*, 2001, pp. 485-488.

[29] R. Farjad-Rad, W. Dally, H.-T. Ng, R. Senthinathan, M.-J. Lee, R. Rathi*, et al.*, "A low-power multiplying DLL for low-jitter multigigahertz clock generation in highly integrated digital chips," *Solid-State Circuits, IEEE Journal of,* vol. 37, pp. 1804-1812, 2002.

[30] S. Gierkink, "An 800MHz-122dBc/Hz-at-200kHz Clock Multiplier based on a Combination of PLL and Recirculating DLL," in *Solid-State Circuits Conference, 2008. ISSCC 2008. Digest of Technical Papers. IEEE International*, 2008, pp. 454-627.

[31] B. M. Helal, M. Z. Straayer, G.-Y. Wei, and M. H. Perrott, "A highly digital MDLL-based clock multiplier that leverages a self-scrambling time-to-digital converter to achieve subpicosecond jitter performance," *Solid-State Circuits, IEEE Journal of,* vol. 43, pp. 855-863, 2008.

[32] K.-J. Hsiao and T.-C. Lee, "The design and analysis of a fully integrated multiplying DLL with adaptive current tuning," *Solid-State Circuits, IEEE Journal of,* vol. 43, pp. 1427-1435, 2008.

[33] A. Elshazly, R. Inti, B. Young, and P. K. Hanumolu, "A 1.5 GHz 890uW digital MDLL with 400fs rms integrated jitter, -55.6 dBc reference spur and 20fs/mV supply-noise sensitivity using 1b TDC," in *Solid-State Circuits Conference Digest of Technical Papers (ISSCC), 2012 IEEE International*, 2012, pp. 242-244.

[34] D. Park and S. Cho, "A 14.2 mW 2.55-to-3GHz cascaded PLL with reference injection, 800MHz delta-sigma modulator and 255fs rms integrated jitter in 0.13 um CMOS," in *Solid-State Circuits*

Conference Digest of Technical Papers (ISSCC), 2012 IEEE International, 2012, pp. 344-346.

[35] A. Elshazly, R. Inti, B. Young, and P. K. Hanumolu, "Clock Multiplication Techniques Using Digital Multiplying Delay-Locked Loops," *Solid-State Circuits, IEEE Journal of,* vol. 48, pp. 1416-1428, 2013.

[36] S. Ye, L. Jansson, and I. Galton, "A multiple-crystal interface PLL with VCO realignment to reduce phase noise," *Solid-State Circuits, IEEE Journal of,* vol. 37, pp. 1795-1803, 2002.

[37] S. Ye and I. Galton, "Techniques for phase noise suppression in recirculating DLLs," *Solid-State Circuits, IEEE Journal of,* vol. 39, pp. 1222-1230, 2004.

[38] C.-F. Liang and K.-J. Hsiao, "An injection-locked ring PLL with self-aligned injection window," in *Solid-State Circuits Conference Digest of Technical Papers (ISSCC), 2011 IEEE International*, 2011, pp. 90-92.

[39] Y.-C. Huang and S.-I. Liu, "A 2.4 GHz sub-harmonically injection-locked PLL with self-calibrated injection timing," in *Solid-State Circuits Conference Digest of Technical Papers (ISSCC), 2012 IEEE International*, 2012, pp. 338-340.

[40] S.-y. Lee, T. Kamimura, S. Yonezawa, A. Shirane, S. Ikeda, H. Ito*, et al.*, "A Multi-Band Quadrature Clock Generator With High-Pass-Filtered Pulse Injection Technique," 2013.

[41] K. Masu, "An Inductorless Cascaded Phase-Locked Loop with Pulse Injection Locking Technique in 90 nm CMOS," *International Journal of Microwave Science and Technology,* vol. 2013, 2013.

[42] P. Madoglio, M. Zanuso, S. Levantino, C. Samori, and A. L. Lacaita, "Quantization effects in all-digital phase-locked loops," *Circuits and Systems II: Express Briefs, IEEE Transactions on,* vol. 54, pp. 1120-1124, 2007.

[43] R. B. Staszewski and P. T. Balsara, *All-digital frequency synthesizer in deep-submicron CMOS*: Wiley. com, 2006.

[44] L. Vercesi, L. Fanori, F. De Bernardinis, A. Liscidini, and R. Castello, "A dither-less all digital PLL for cellular transmitters," *Solid-State Circuits, IEEE Journal of,* vol. 47, pp. 1908-1920, 2012.

[45] A. A. Abidi, "Phase noise and jitter in CMOS ring oscillators," *Solid-State Circuits, IEEE Journal of,* vol. 41, pp. 1803-1816, 2006.

[46] L. Xiu, "A Flying-Adder On-Chip Frequency Generator for Complex SoC Environment," *Circuits and Systems II: Express Briefs, IEEE Transactions on,* vol. 54, pp. 1067-1071, 2007.

[47] L. Xiu, "A flying-adder PLL technique enabling novel approaches for video/graphic applications," *Consumer Electronics, IEEE Transactions on,* vol. 54, pp. 591-599, 2008.

[48] L. Xiu, W. Li, J. Meiners, and R. Padakanti, "A novel all-digital PLL with software adaptive filter," *Solid-State Circuits, IEEE Journal of,* vol. 39, pp. 476-483, 2004.

[49] L. Xiu, W.-T. Lin, and T.-T. Lee, "Flying-Adder Fractional Divider Based Integer-N PLL: 2nd Generation FAPLL as On-Chip Frequency Generator for SoC," 2013.

[50] A. V. Oppenheim, R. W. Schafer, and J. R. Buck, *Discrete-time signal processing* vol. 5: Prentice Hall Upper Saddle River, 1999.

[51] J. W. Rogers, C. Plett, and F. Dai, *Integrated circuit design for high-speed frequency synthesis*: Artech House Boston, London, 2006.

[52] V. Kratyuk, P. K. Hanumolu, U.-K. Moon, and K. Mayaram, "A design procedure for all-digital phase-locked loops based on a charge-pump phase-locked-loop analogy," *Circuits and Systems II: Express Briefs, IEEE Transactions on,* vol. 54, pp. 247-251, 2007.

[53] M. H. Perrott, "Tutorial on Digital Phase-Locked Loops," in *Custom Integrated Circuits Conference*, 2009.

[54] C.-M. Hsu, "Techniques for high-performance digital frequency synthesis and phase control," Massachusetts Institute of Technology, 2008.

[55] B. Miller and B. Conley, "A multiple modulator fractional divider," in *Frequency Control, 1990., Proceedings of the 44th Annual Symposium on*, 1990, pp. 559-568.

[56] 이형록, "다중위상 클럭을 이용한 저 잡음 주파수 합성기의 설계," 공학박사, 전기·컴퓨터공학부, 서울대학교 대학원, 2006.

[57] VESA, "vesa and industry standards and guidelines for computer display monitor timing (DMT)," Version 1.0, Revision 11 ed, 2007.

[58] C.-C. Chung and C.-Y. Ko, "A fast phase tracking ADPLL for video pixel clock generation in 65 nm CMOS technology," *Solid-State Circuits, IEEE Journal of,* vol. 46, pp. 2300-2311, 2011.

[59] H. Marie and P. Belin, "R, G, B acquisition interface with line-locked clock generator for flat panel display," *Solid-State Circuits, IEEE Journal of,* vol. 33, pp. 1009-1013, 1998.

[60] G.-j. Xie and C. Wang, "An all-digital PLL for video pixel clock regeneration applications," in *Computer Science and Information Engineering, 2009 WRI World Congress on*, 2009, pp. 392-396.

[61] C. Lahuec, J. Horan, and J. Duigan, "Programmable video clock

synthesizer with sub 0.5 ns drift," in *Circuits and Systems, 2002. ISCAS 2002. IEEE International Symposium on*, 2002, pp. IV-783-IV-786 vol. 4.

[62] M. Song, Y.-H. Kwak, S. Ahn, W. Kim, B. Park, and C. Kim, "A 10mhz to 315mhz cascaded hybrid pll with piecewise linear calibrated tdc," in *Custom Integrated Circuits Conference, 2009. CICC'09. IEEE*, 2009, pp. 243-246.

[63] H.-R. Lee, O. Kim, K. Jung, J. Shin, and D.-K. Jeong, "A PVT-Tolerant Low-1/f Noise Dual-Loop Hybrid PLL in 0.18/spl mu/m," in *Solid-State Circuits Conference, 2006. ISSCC 2006. Digest of Technical Papers. IEEE International*, 2006, pp. 2402-2411.

[64] D. E. Calbaza, I. Cordos, N. Seth-Smith, and Y. Savaria, "An ADPLL circuit using a DDPS for genlock applications," in *Circuits and Systems, 2004. ISCAS'04. Proceedings of the 2004 International Symposium on*, 2004, pp. IV-569-72 Vol. 4.

[65] W. Kim, J. Park, J. Kim, T. Kim, H. Park, and D. Jeong, "A 0.032 mm 2 3.1 mW synthesized pixel clock generator with 30ps rms integrated jitter and 10-to-630MHz DCO tuning range," in *Solid-State Circuits Conference Digest of Technical Papers (ISSCC), 2013 IEEE International*, 2013, pp. 250-251.

[66] R. B. Staszewski, I. Bashir, and K. Waheed, "Dynamic bandwidth adjustment of an RF all-digital PLL," in *Radio Frequency Integrated Circuits Symposium (RFIC), 2011 IEEE*, 2011, pp. 1-4.

[67] D.-H. Oh, K.-J. Choo, and D.-K. Jeong, "Phaseçĺôfrequency detecting time-to-digital converter," *Electronics letters,* vol. 45, pp. 201-202, 2009.

[68] 오도환, "A study on design of all-digital phase-locked loop," 공학박사, 전기· 컴퓨터공학부, 서울대학교 대학원, 2009.

[69] M. Z. Straayer and M. H. Perrott, "An efficient high-resolution 11-bit noise-shaping multipath gated ring oscillator TDC," in *VLSI Circuits, 2008 IEEE Symposium on*, 2008, pp. 82-83.

[70] D. Sheng, C.-C. Chung, and C.-Y. Lee, "An ultra-low-power and portable digitally controlled oscillator for SoC applications," *Circuits and Systems II: Express Briefs, IEEE Transactions on,* vol. 54, pp. 954-958, 2007.

[71] C.-T. Wu, W.-C. Shen, W. Wang, and A.-Y. Wu, "A two-cycle lock-in time ADPLL design based on a frequency estimation algorithm," *Circuits and Systems II: Express Briefs, IEEE Transactions on,* vol. 57, pp. 430-434, 2010.

[72] Y. Park and D. D. Wentzloff, "An all-digital PLL synthesized from a digital standard cell library in 65nm CMOS," in *Custom Integrated*

Circuits Conference (CICC), 2011 IEEE, 2011, pp. 1-4.

[73] M. Maymandi-Nejad and M. Sachdev, "A monotonic digitally controlled delay element," *Solid-State Circuits, IEEE Journal of,* vol. 40, pp. 2212-2219, 2005.

[74] JitterTime. *Converting Phase Noise (dBc/Hz) to Phase Jitter (ps RMS)*. Available: http://www.jittertime.com/articles/pnsheet.shtml

초록

트랜지스터의 소형화(Scaling)가 진행됨에 따라서 위상동기루프(PLL)를 설계함에 있어서 많은 도전에 직면하고 있다. 특히나 루프필터에서의 누설전류와 동작전압이 낮아짐에 따른 동작영역의 제한은 기존의 아날로그 방식의 회로기술을 최신 공정기술에 적용하기 힘들게 하고 있다. 이를 해결하기 위하여 디지털 위상동기루프(All Digital PLL) 기술이 최근 대체 기술로 많은 연구가 이루어 지고 있다. 디지털 PLL은 전체회로를 디지털 회로로 대체함으로써 기존의 아날로그 PLL에서 문제가 되었던, 누설전류 와 동작영역 제한 문제를 해결하였다. 하지만 낮은 동작전압에서 원하는 Jitter 성능을 달성하는 것은 여전히 문제로 남아있다. 본 논문에서는 저 전압에서도 우수한 Jitter 특성을 달성하기 위하여 이중루프 (Dual loop) 구조를 갖는 PLL을 구현하였다. 또한 높은 해상도와 넓은 동작영역을 만족하는 DCO와 TDC를 구현하기 위하여, 상향식 (Bottom-Up) 다단계 제어방식 (Multi-step control)을 제안하였다. 설계방법 측면에서 기존에는 전체회로를 HDL(Hardware Description Language)로 기술하고 있음에도, chip을 구현함에 있어서 많은 부분을 설계자의 custom 설계에 의존하여 왔다. 본 논문 에서는 전체회로를 자동으로 합성하기 위하여 새로운 단위 Cell layout 기법을 이용한 설계 방법을 제안하였다. 제안된 기술을 사용함으로써, 자동으로 배치 및 배선 (Place and routing)를 진행했음에도 선형 성(Linearity) 열화가 없었다. 제안된 이중루프 PLL (Dual Loop PLL) 구조를 이용하여 픽셀 클락 생성 기 (Pixel clock generator)를 구

현하였다. 전체회로는 제안된 단위 셀 (Unit cell) 기반 레이아웃 기법을 바탕으로, 28nm CMOS 공정기술을 이용하여 자동 합성 되었다. 이중 루프 PLL의 지터(Jitter)성능을 최적화 하기 위하여, 선형 노이즈 모델을 이용하여 전체 루프를 최적화 하였다. 테스트 칩은 $0.032mm^2$ 의 면적을 가지며, 100 kHz 의 매우 낮은 입력 주파수 신호에 대해서 15ps_rms의 낮은 누적 지터 (Integrated Jitter)를 달성하였다. 전체 회로는 1.0V 전압으로 동작하며, 3.1mW의 낮은 전력을 소비 하였다.

키워드 : 위상동기루프, 셀 기반, 합성, 지터, 픽셀 클락, 이중루프.
학번 : 2010-30216

To my family